



Universidad
Carlos III de Madrid

Departamento de Informática
Ingeniería en Informática

PROYECTO FIN DE CARRERA

Infraestructura móvil para el despliegue de servicios basados en localización en la Universidad Carlos III de Madrid (Plataforma Android)

Autor: Javier Sánchez Hernández

Tutor: Dr. Javier García Guzmán

Leganés, julio de 2012

Título: INFRAESTRUCTURA MÓVIL PARA EL DESPLIEGUE DE SERVICIOS BASADOS EN LA LOCALIZACIÓN EN LA UNIVERSIDAD CARLOS III DE MADRID (PLATAFORMA ANDROID).

Autor: Javier Sánchez Hernández

Director: Dr. Javier García Guzmán

EL TRIBUNAL

Presidente: Dr. Antonio de Amescua Seco

Vocal: María Isabel Barro Vivero

Secretario: Dr. Alberto Heredia García

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 18 de julio de 2012 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Cuando se acerca el final de la carrera y uno presenta su proyecto no se puede dejar de volver la vista atrás y pensar en estos años en los que he aprendido tantas cosas, gracias a los profesores que he tenido en las diferentes asignaturas, profesores que han compartido sus conocimientos y experiencias de trabajo con nosotros que nos han servido para crecer y aprender junto a ellos a ser mejores profesionales.

El ambiente en la Universidad Carlos III de Madrid, la cercanía de los docentes y la calidad de la enseñanza han sido para mí algo fundamental en mi vida.

Tengo que agradecer también todo el apoyo de mi familia, de mis padres, Luis y Julia, de mis hermanas, Ana y a Carmen. Ellos siempre han confiado en mí y me han transmitido valores gracias a los cuales he podido llegar a este punto, un punto y aparte en la vida. Vida que estoy comenzando como profesional.

Quiero agradecer también su apoyo a mi tutor, Javier García Guzmán. Él me ha guiado durante el desarrollo de este proyecto, ayudándome a buscar la excelencia en el trabajo. No quiero dejar de mencionar a mis compañeros del grupo SEL-UC3M, en especial a Juan Carlos por su apoyo y consejo a la hora de realizar la aplicación.

Finalmente quiero recordar a mis compañeros de carrera y compañeros de prácticas, Alberto Sánchez, Alejandro, Isabel, Irene, Alberto López, Héctor y muchos otros con los que he compartido tantas horas, esfuerzos y alegrías por el trabajo bien hecho.

“Hazlo todo tan simple como sea posible, pero no más simple”

Albert Einstein

Resumen

Los servicios basados en localización ofrecen al usuario un servicio personalizado basado en la ubicación geográfica. Estos servicios suelen consistir en mapas sobre los que se muestra información útil para el usuario.

El problema que resuelve este proyecto es la dificultad que tienen algunas personas para orientarse cuando buscan algo en lugares desconocidos. En este caso concreto, los campus de la Universidad Carlos III de Madrid.

Este problema junto con la importancia que está alcanzando el uso de las tecnologías móviles combinado con la geolocalización ya sea mediante satélites o redes inalámbricas, es lo que motiva la realización de este PFC.

El objetivo principal de este proyecto es desarrollar un servicio que permita al usuario encontrar lugares y personas dentro de los campus de la universidad ya que en ocasiones es complicado encontrar algunas aulas. Los objetivos secundarios consisten en realizar ciertos módulos que combinados conforman un sistema capaz de proporcionar servicios de búsqueda geolocalizada.

La metodología de trabajo que se ha utilizado durante la realización del proyecto ha sido una metodología ágil de desarrollo: Scrum. Se ha elegido esta opción porque se adaptaba bien a la planificación adaptativa que hemos seguido.

Se ha elegido la plataforma Android como plataforma de desarrollo, y se han utilizado herramientas para el desarrollo en Android, como son el lenguaje de programación Java junto con ficheros XML y la herramienta Eclipse. También es reseñable que para todas las funcionalidades se han realizado pruebas de sistema, por lo que se asegura el correcto funcionamiento de la aplicación.

Como resultado, se ha obtenido una aplicación móvil que ofrece al usuario servicios de búsqueda geolocalizada de lugares y personas dentro de los campus universitarios y la capacidad de orientarse gracias a la utilización de los planos del interior de los edificios superpuestos. Además ofrece una vista de detalle para cada lugar y persona que da la posibilidad de realizar ciertas acciones, la posibilidad de configurar el tipo de búsqueda y un tutorial dirigido a los nuevos usuarios. La aplicación está disponible a través de Google Play desde el mes de abril.

Abstract

Location-based services offer users a personalized service based on geographic location. These services typically include maps that show useful information for the user.

The problem solved by this project is the difficulty to get their bearings that some people have when they search for something in unfamiliar places. In this case, the different campuses of the Carlos III University of Madrid.

This problem along with the importance that is reaching the use of mobile technologies combined with geolocation either provided by satellite or by wireless networks is what motivates the realization of the project.

The main objective of this project is to develop a service that allows users to find places and people within the university campuses as it is sometimes difficult to find some classrooms. Secondary objectives consist of implement certain modules that combine to form a system able to provide geolocated search services.

The working methodology used during the project implementation has been an agile development methodology: Scrum. We have chosen this option because it was well suited to the adaptive planning that we have followed.

We chose Android as development platform and we have used the tools for Android development such as the Java programming language with XML files and the Eclipse tool. It is also noteworthy that for all functions, we have made system tests to ensure the proper functioning of the application.

As a result, we obtained a mobile application that provides users with geolocation-based search services of places and people within the university campuses and the ability to get their bearings through the use of indoor maps overlaid on the main map. It also offers a detail view for each place and person that makes possible to perform certain actions, the ability to configure the search type and a tutorial aimed to new users. The application is available through Google Play from April.

Índice General

CAPÍTULO 1 INTRODUCCIÓN	1
1.1 PROBLEMA	2
1.2 MOTIVACIÓN DEL PROYECTO	2
1.3 OBJETIVOS	3
1.4 MÉTODO DE RESOLUCIÓN	4
1.5 TERMINOLOGÍA.....	4
1.6 CONTENIDO DE LA MEMORIA.....	6
CAPÍTULO 2 ESTADO DEL ARTE.....	7
2.1 ANÁLISIS DE APLICACIONES MÓVILES Y WEB RELACIONADAS CON LOS OBJETIVOS ESTABLECIDOS ...	8
2.2 ENTORNO DE DESARROLLO MÓVIL SELECCIONADO.....	15
2.3 DESARROLLO ÁGIL COMO MÉTODO PARA REALIZAR EL PROYECTO	21
2.4 CONCLUSIONES DEL ANÁLISIS.....	25
CAPÍTULO 3 ESPECIFICACIÓN DE REQUISITOS.....	27
3.1 INTRODUCCIÓN	28
3.2 DESCRIPCIÓN GENERAL DEL SISTEMA A DESARROLLAR	30
3.3 REQUISITOS ESPECÍFICOS.....	32
CAPÍTULO 4 DISEÑO DE LA APLICACIÓN	47
4.1 PAPER PROTOTYPE.....	48
4.2 ALTERNATIVAS DE DISEÑO	58
4.3 DIAGRAMA DE CLASES	64
4.4 DIAGRAMAS DE SECUENCIA.....	69
4.5 DISEÑO DE LA BASE DE DATOS	75
4.6 DISEÑO DEL REPOSITORIO DE PLANOS.....	76
CAPÍTULO 5 PRUEBAS DE LA APLICACIÓN	77
5.1 PRUEBAS UNITARIAS	78
5.2 PRUEBAS DE INTEGRACIÓN	85
5.3 PRUEBAS DE SISTEMA.....	85
CAPÍTULO 6 CONCLUSIONES Y LÍNEAS FUTURAS.....	97
6.1 CONCLUSIONES.....	98
6.2 LÍNEAS FUTURAS	100
BIBLIOGRAFÍA.....	103
APÉNDICE A PRESUPUESTO DEL PROYECTO.....	107
APÉNDICE B MANUAL DE USUARIO.....	111

Lista de Figuras

FIGURA 2.1: GOOGLE MAPS 6.0	9
FIGURA 2.2: POINT INSIDE 3.1.0	10
FIGURA 2.3: APLICACIÓN DE EJEMPLO UTILIZANDO INDOOR MAPS AND POSITIONING ...	11
FIGURA 2.4: UNIV. MURCIA 2.3	12
FIGURA 2.5: ALABAMA 1.5.2	12
FIGURA 2.6: BALL STATE MAP APPLICATION 1.0.....	13
FIGURA 2.7: MIT MOBILE 2.3.1 (VISTA DE MAPA)	14
FIGURA 2.8: MIT MOBILE 2.3.1 (VISTA DE DETALLE)	14
FIGURA 2.9: DIVISIÓN DEL MERCADO DE S.O. MÓVILES	16
FIGURA 2.10: VERSIONES DE ANDROID FUNCIONANDO ACTUALMENTE.....	18
FIGURA 2.11: ARQUITECTURA DEL SISTEMA ANDROID	19
FIGURA 2.12: DIAGRAMA DE FLUJO DE SCRUM.....	24
FIGURA 4.1: VISTA PRINCIPAL.....	49
FIGURA 4.2: VISTA DE CONFIGURACIÓN	49
FIGURA 4.3: VISTA DE LUGAR	50
FIGURA 4.4: VISTA AMPLIADA	50
FIGURA 4.5: PANTALLA PRINCIPAL.....	51
FIGURA 4.6: INTRODUCCIÓN DE TÉRMINO DE BÚSQUEDA	52
FIGURA 4.7: RESULTADO DE BÚSQUEDA DE LUGAR.....	53
FIGURA 4.8: DETALLES DE LUGAR.....	53
FIGURA 4.9: RESULTADO DE BÚSQUEDA DE PERSONA	54
FIGURA 4.10: DETALLES DE PERSONA.....	55
FIGURA 4.11: OPCIONES DEL MENÚ PRINCIPAL.....	56
FIGURA 4.12: PANTALLA DE CONFIGURACIÓN.....	57
FIGURA 4.13: VISTA DE ACERCA DE.....	58
FIGURA 4.14: ALTERNATIVA DE DISEÑO 1	62
FIGURA 4.15: ALTERNATIVA DE DISEÑO 2	63
FIGURA 4.16: ALTERNATIVA DE DISEÑO 3	64
FIGURA 4.17: DIAGRAMA DE COMPONENTES.....	65
FIGURA 4.18: MODEL	66
FIGURA 4.19: LISTS	66
FIGURA 4.20: CONTROLLER.....	67
FIGURA 4.21: MAPAS	68
FIGURA 4.22: VIEWS	68
FIGURA 4.23: OVERLAYS.....	69
FIGURA 4.24: INICIO DE LA APLICACIÓN	70
FIGURA 4.25: COMPROBACIÓN DE LA CONEXIÓN A INTERNET	71
FIGURA 4.26: BÚSQUEDA DE UN LUGAR.....	72
FIGURA 4.27: MODIFICACIÓN EN LA CONFIGURACIÓN	73

FIGURA 4.28: VISTA DE DETALLE DE PERSONA	74
FIGURA 4.29: DIAGRAMA ENTIDAD-RELACIÓN	75
FIGURA 4.30: DIAGRAMA RELACIONAL	76
FIGURA 5.1: RESULTADOS DE NETWORKTESTERTEST	79
FIGURA 5.2: RESULTADOS DE STRINGHANDLERTEST	79
FIGURA 5.3: RESULTADOS SEARCHTASKTEST	80
FIGURA 5.4: RESULTADOS DE PERSONTEST	81
FIGURA 5.5: RESULTADOS DE PLACETEST	81
FIGURA 5.6: RESULTADOS DE UC3MMAPASACTIVITYTEST	83
FIGURA 5.7: RESULTADOS DE ABOUTACTIVITYTEST	84
FIGURA 5.8: RESULTADOS DE LISTITEMTEST	84
FIGURA 6.1: MENÚ DE OPCIONES	112
FIGURA 6.2: CONFIGURACIÓN DE PREFERENCIAS	112
FIGURA 6.3: BÚSQUEDA DE UN LUGAR	113
FIGURA 6.4: RESULTADO DE BÚSQUEDA DE UN LUGAR	113
FIGURA 6.5: BÚSQUEDA DE UNA PERSONA	114
FIGURA 6.6: RESULTADO DE BÚSQUEDA DE UNA PERSONA	114
FIGURA 6.7: DETALLE DE UNA PERSONA	115
FIGURA 6.8: OPCIONES DE PERSONA	115

Lista de Tablas

TABLA 2.1: COMPARATIVA ENTRE APLICACIONES	15
TABLA 2.2: NOMENCLATURA DE LAS VERSIONES DE ANDROID	17
TABLA 2.3: METODOLOGÍAS ÁGILES VS. TRADICIONALES	23
TABLA 3.1: IDENTIFICADOR - REQUISITO DE EJEMPLO	33
TABLA 3.2: RF-001 - SELECCIÓN DEL TIPO DE BÚSQUEDA	34
TABLA 3.3: RF-002 - BUSCAR SEGÚN UN TÉRMINO DE BÚSQUEDA	34
TABLA 3.4: RF-003 - MOSTRAR DETALLES DEL LUGAR	35
TABLA 3.5: RF-004 - MOSTRAR DETALLES DE LA PERSONA	35
TABLA 3.6: RF-005 - REPORTAR INCIDENCIA	36
TABLA 3.7: RF-006 - ENVIAR CORREO ELECTRÓNICO A LA PERSONA	36
TABLA 3.8: RF-007 - LLAMAR A LA PERSONA	37
TABLA 3.9: RF-008 - AGREGAR A LA PERSONA COMO CONTACTO	37
TABLA 3.10: RF-009 - MOSTRAR EL TWITTER DEL SEL	37
TABLA 3.11: RF-010 - COMPARTIR LA APLICACIÓN EN REDES SOCIALES	38
TABLA 3.12: RF-011 - MOSTRAR LOS CRÉDITOS	38
TABLA 3.13: RF-012 - SELECCIONAR CAMPUS	38
TABLA 3.14: RF-013 - SELECCIONAR TIPO DE PLANO	39
TABLA 3.15: RF-014 - TRADUCCIÓN AL INGLÉS	39
TABLA 3.16: RI-001 - API DE COMUNICACIÓN	40
TABLA 3.17: RI-002 - COMUNICACIÓN A TRAVÉS DE URL	40
TABLA 3.18: RI-003 - VISTA DE BÚSQUEDA	41
TABLA 3.19: RI-004 - VISTA DE CONFIGURACIÓN	41
TABLA 3.20: RI-005 - VISTA DE ACERCA DE	42
TABLA 3.21: RI-006 - VISTA DE CRÉDITOS	42
TABLA 3.22: RI-008 - VISTA DE AYUDA	43
TABLA 3.23: RI-008 – COMUNICACIÓN CON EL SERVICIO WEB	43
TABLA 3.24: RR-001 - TIEMPO MÁXIMO EN LAS CONSULTAS	44
TABLA 3.25: RR-002 - ABRIR NAVEGADOR EN REDES PÚBLICAS	44
TABLA 3.26: RD-001 - VERSIÓN DEL SISTEMA OPERATIVO	45
TABLA 3.27: RD-002 - VERSIÓN DEL SDK DE JAVA	45
TABLA 5.1: PLANTILLA DE PRUEBA	86
TABLA 5.2: P1.1 - LA APLICACIÓN DEBERÁ INICIARSE	86
TABLA 5.3: P1.2 - RECIBIR PARÁMETROS DESDE OTRA APLICACIÓN	87
TABLA 5.4: P1.3 - RECIBIR PARÁMETROS DESDE UNA URL	87
TABLA 5.5: P2.1 - BÚSQUEDA DE UN LUGAR	88
TABLA 5.6: P2.2 INCIDENCIA DE LUGAR	89
TABLA 5.7: P2.3 BÚSQUEDA DE UNA PERSONA	90
TABLA 5.8: P2.4 FUNCIONALIDAD DE LA VISTA DE DETALLE DE PERSONA	92
TABLA 5.9: P2.5 TIPO DE MAPA	92

TABLA 5.10: P3.1 TIEMPO MÁXIMO DE RESPUESTA	93
TABLA 5.11: P3.2 RED INALÁMBRICA PÚBLICA	93
TABLA 5.12: P4.1 ACERCA DE	94
TABLA 5.13: P5.1 AYUDA AL USUARIO	95
TABLA 6.1: COSTE ASOCIADO AL PERSONAL.....	108
TABLA 6.2: COSTE DE AMORTIZACIONES	109
TABLA 6.3: COSTE DE LICENCIA	109
TABLA 6.4: COSTE TOTAL.....	110

CAPÍTULO 1 INTRODUCCIÓN

1.1	PROBLEMA	2
1.2	MOTIVACIÓN DEL PROYECTO.....	2
1.3	OBJETIVOS.....	3
1.4	MÉTODO DE RESOLUCIÓN	4
1.5	TERMINOLOGÍA.....	4
1.5.1	GLOSARIO DE TÉRMINOS	4
1.5.2	ABREVIATURAS.....	5
1.6	CONTENIDO DE LA MEMORIA	6

En este capítulo se hace una breve introducción al presente proyecto, exponiendo en primer lugar cuál es el problema a resolver, a continuación se expone cuál es su motivación, cuáles son sus objetivos, el método de resolución utilizado para llevarlo a cabo, la terminología que se utiliza en esta memoria y por último, cuál es el contenido ofrecido en esta memoria.

1.1 PROBLEMA

El problema que se ha detectado y que se pretende resolver mediante este proyecto consiste en la dificultad que tienen algunas personas para encontrar lugares dentro de los campus de la Universidad Carlos III de Madrid (UC3M) sabiendo únicamente algunos datos. En la mayor parte de casos, el único dato que se sabe es el identificador de la sala a la que se desea llegar, pero a las personas que no están familiarizadas con el sistema utilizado, les resulta complicado interpretar ese identificador.

Este problema afecta sobre todo a nuevos estudiantes que todavía no conocen los campus, a miembros de la comunidad universitaria que por alguna razón tienen que visitar otro campus que no conocen bien, a estudiantes que el día del examen no saben bien a qué aula dirigirse porque no está en su edificio habitual, a visitantes que se acercan a la universidad para asistir a reuniones, congresos o actividades varias como pueden ser espectáculos en el auditorio o graduaciones de estudiantes.

A veces se utilizan ciertos términos para denominar puntos geográficos dentro de los campus que una persona ajena puede no conocer, como puede ser la locomotora o el cuerno que se encuentran en el campus de Leganés. También ocurre esto con lugares menos extraños pero a su vez difíciles de encontrar como puede ser la cafetería del edificio Padre Soler, la conserjería del edificio Torres Quevedo o la oficina bancaria del edificio Agustín de Betancourt.

Otro problema común consiste en encontrar un despacho cuando no se sabe cuál es el identificador. Hay personas que tienen dificultad para recordar ciertas estructuras como son los identificadores de despacho, pero por el contrario, se les da bien recordar nombres. El ejemplo para este caso sería un alumno que desea ir a una tutoría con su profesor de cierta asignatura y no recuerda cuál es su despacho pero sí recuerda su nombre, sus apellidos o incluso su correo electrónico, ya que la parte inicial (hasta la arroba) sirve como término de búsqueda.

Esto mismo ocurre cuando un docente desea encontrar el despacho de un compañero y no recuerda el identificador de despacho o cuando cualquier miembro de la comunidad universitaria busca un servicio como puede ser el Punto de Información del Campus (PIC).

1.2 MOTIVACIÓN DEL PROYECTO

El problema que se describe en el apartado anterior y el afán por solucionar los problemas de las personas que nos rodean, en este caso los miembros de la comunidad universitaria, es lo que nos motiva a pensar en una solución que permita, a las personas que lo deseen, encontrar los lugares y personas que buscan dentro de los distintos campus de la universidad.

Este proyecto también está motivado por la importancia que están alcanzando las tecnologías móviles en los últimos años y por el interés del grupo de investigación SEL-UC3M y la Universidad Carlos III de Madrid en desarrollar aplicaciones móviles que faciliten la vida universitaria a los miembros de la comunidad universitaria, así como a los visitantes que no siendo parte de esta comunidad sí puedan disfrutar de estos avances tecnológicos.

Por tanto, el propósito de este proyecto de fin de carrera consiste en desarrollar una serie de facilidades que se describen a continuación:

- Una base de datos geográfica en la que se georreferenciarán todas las aulas y despachos, auditorios, salas de reuniones y puntos de interés dentro de los diferentes campus de la Universidad Carlos III de Madrid de manera que los usuarios tengan la posibilidad de encontrar el mayor número de lugares y personas sin necesidad de esforzarse.
- Un repositorio de mapas que tiene como objetivo mostrar al usuario los planos de los edificios impresos sobre el mapa de la universidad. Esta característica hace que sea más sencillo llegar al lugar deseado ya que en los planos se muestran los accesos a los pasillos y escaleras con colores fáciles de interpretar.
- Una aplicación móvil en plataforma Android para la provisión de servicios básicos de localización para la Universidad Carlos III de Madrid, incluyendo búsqueda de lugares y personas en todos los campus de la universidad. También se ofrecerá información acerca del lugar o la persona buscados y la posibilidad de interactuar con ella.

Otra de las situaciones que me llevan a realizar este proyecto es la disponibilidad de un laboratorio del grupo de investigación SEL-UC3M en el que desarrollar todas estas facilidades, de manera que los medios para realizar el proyecto sean los adecuados y la comunicación con mi tutor sea fluida. También con esta situación tengo mayor facilidad de comunicación con Juan Carlos Alonso Durán que es el responsable del desarrollo de la parte del servidor con la que se comunica la aplicación.

1.3 OBJETIVOS

El principal objetivo de este proyecto es **desarrollar servicios que faciliten la orientación a los usuarios dentro de los diferentes campus de la Universidad Carlos III de Madrid (UC3M).**

Además de este objetivo principal, este proyecto tiene otros objetivos que se enumeran a continuación:

- **Desarrollo de un servicio de localización para la UC3M.** En primer lugar, se deberá poner en marcha un servicio Web que permita georreferenciar lugares dentro de los campus de la UC3M. También se podrán realizar consultas georreferenciadas. Para ello, debe estar sincronizado con el directorio de PAS y PDI de la UC3M.

- **Georreferenciación de los campus de la UC3M.** El siguiente paso para realizar el proyecto consiste en georreferenciar todas las aulas, despachos y puntos de interés en cada uno de los campus.
- **Adaptación de los planos de los edificios.** Complementario al trabajo de georreferenciación, se encuentra el de adaptación de los planos de la universidad de manera que se puedan mostrar superpuestos en un mapa de Google.
- **Desarrollo de una aplicación Android.** Como punto final al proyecto, se ha desarrollado una aplicación Android que utiliza los servicios desplegados al cumplir los anteriores objetivos y que permite buscar personas y lugares.

1.4 MÉTODO DE RESOLUCIÓN

El método llevado a cabo para la resolución del problema ha sido iterativo, ya que en un principio no se tenían claros los requisitos que debía cumplir el sistema y se pretendía definirlos a lo largo del desarrollo del proyecto, dependiendo de los avances y opciones que fueran surgiendo.

Las iteraciones que se plantearon desde un inicio de este proyecto están basadas en las que se proponen en algunas de las metodologías ágiles, en las que las reuniones del equipo de desarrollo toman una importancia considerable. La frecuencia de las reuniones de se estableció al inicio del proyecto en dos semanas, tiempo en el que los desarrolladores implementaban los requisitos estipulados para revisarlos en la siguiente reunión.

En cada una de las iteraciones en primer lugar había una reunión con todo el grupo de desarrollo para evaluar el prototipo de la iteración anterior (si lo había) de manera que el cliente pudiera tener software funcionando al final de cada una de las iteraciones y definir los requisitos que el sistema debía cumplir en la siguiente reunión.

Al finalizar las reuniones, se ha proporcionado a los desarrolladores una lista de tareas a realizar en un tiempo estipulado, de manera que se pueda verificar qué objetivos se han cumplido y cuáles no.

Para poder controlar todas estas tareas se ha hecho uso de la herramienta *Redmine*, mediante de la cual es posible crear tareas y asignárselas al desarrollador que debe encargarse de realizarlas. En ella, los desarrolladores pueden indicar qué porcentaje han realizado de cada tarea e indicar el número de horas utilizado para ello.

1.5 TERMINOLOGÍA

En este apartado se recoge un glosario en el que se explican brevemente algunos de los términos utilizados en este documento con el fin de que el lector sea capaz de comprender el total del contenido.

1.5.1 GLOSARIO DE TÉRMINOS

App es la forma de denominar a las aplicaciones informáticas para dispositivos móviles como son los Smartphone o las tabletas.

Geolocalización o georreferenciación es un concepto que hace referencia a la identificación de una situación geográfica de manera automática.

iOS es el sistema operativo móvil de Apple desarrollado originalmente para el iPhone y siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV.

Smartphone (teléfono inteligente) es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de computación y conectividad que un teléfono móvil convencional.

Tablet PC o tableta ('ordenador personal en tableta') es una computadora portátil con la que se puede interactuar a través de una pantalla táctil o multitáctil. Para trabajar con la computadora, el usuario puede utilizar una pluma stylus o los dedos, sin necesidad de teclado físico ni ratón.

Wifi Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Generalmente se utiliza para acceder a Internet a través de un router con esta tecnología.

1.5.2 ABREVIATURAS

ADB Del inglés Android Debug Bridge, Puente de Depuración Android. Es una herramienta versátil de línea de comandos que permite comunicar el Entorno de Desarrollo Integrado con una instancia del emulador o con un dispositivo Android conectado.

API Del inglés *Application Programming Interface*, Interfaz de programación de aplicaciones. Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.

AVD Del inglés *Android Virtual Device*, Dispositivo virtual de Android. Es la forma de llamar al emulador de Android que se puede ejecutar sobre otro SO como puede ser *Windows*, *Linux* o *MacOSX*.

CPU Del inglés *Central Processing Unit*, Unidad central de procesamiento (UCP). Es el componente del computador y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos.

GPS Del inglés *Global Positioning System*, Sistema de Posicionamiento Global. Es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto con una precisión hasta de centímetros, aunque lo habitual son unos pocos metros de precisión.

LDAP Del inglés *Lightweight Directory Access Protocol*, Protocolo Ligero de Acceso a Directorios. Es un protocolo que permite acceder a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

PAS Personal de Administración y Servicios.

PDI Personal Docente e Investigador.

PFC Proyecto Fin de Carrera.

SDK Del inglés *Software Development Kit*, Kit de desarrollo de software. Es un conjunto de bibliotecas que ofrecen al desarrollador unas funcionalidades básicas que permiten crear programas con mayor facilidad.

SO Sistema Operativo.

XML eXtended Markup Language. Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

1.6 CONTENIDO DE LA MEMORIA

En este apartado se explica brevemente el contenido de cada una de las secciones contenidas en esta memoria.

En el capítulo 1, *Introducción*, se expone una motivación del presente proyecto, los objetivos que persigue, el método de resolución del mismo, algunas aclaraciones sobre la terminología y por último una visión general de los contenidos.

En el capítulo 2, *Estado del arte*, se analizan aplicaciones informáticas cuyos objetivos son similares a los expuestos en el apartado 1.3, se expone información acerca del entorno de desarrollo móvil elegido y del método de desarrollo elegido.

En el capítulo 3, *Especificación de requisitos*, se exponen, organizados por tipo, los requisitos que debe cumplir el sistema a implementar.

En el capítulo 4, *Diseño de la aplicación*, se presentan los *paper prototypes* (prototipos en papel) de la aplicación móvil, un diagrama de clases y algunos diagramas de secuencia.

En el capítulo 5, *Pruebas de la aplicación*, se exponen los resultados de las pruebas unitarias, de integración y de sistema.

En el capítulo 6, *Conclusiones*, se presentan las conclusiones a las que se ha llegado durante la realización del proyecto.

Por último, se añaden a la memoria una serie de apéndices con el presupuesto del proyecto y las referencias bibliográficas utilizadas para la realización de este proyecto.

CAPÍTULO 2 ESTADO DEL ARTE

2.1 ANÁLISIS DE APLICACIONES MÓVILES Y WEB RELACIONADAS CON LOS OBJETIVOS ESTABLECIDOS	8
2.1.1 GOOGLE MAPS	9
2.1.2 POINT INSIDE SHOPPING & TRAVEL	10
2.1.3 INDOOR MAPS AND POSITIONING	10
2.1.4 NAO CAMPUS	11
2.1.5 UNIVERSIDAD DE MURCIA	11
2.1.6 UNIVERSITY OF ALABAMA	12
2.1.7 BALL STATE UNIVERSITY MAP APP	13
2.1.8 MIT MOBILE	13
2.2 ENTORNO DE DESARROLLO MÓVIL SELECCIONADO	15
2.2.1 ENTORNOS MÓVILES	15
2.2.2 VISIÓN GENERAL SOBRE ANDROID	16
2.2.3 ESTRUCTURAL DEL SISTEMA ANDROID	18
2.2.4 ENTORNO DE DESARROLLO INTEGRADO	21
2.3 DESARROLLO ÁGIL COMO MÉTODO PARA REALIZAR EL PROYECTO	21
2.3.1 THE AGILE ALLIANCE	22
2.3.2 SCRUM	23
2.4 CONCLUSIONES DEL ANÁLISIS.....	25

El primer paso que se debe dar ante un proyecto de estas características consiste en analizar el contexto en el que se va a desarrollar la aplicación y las aplicaciones ya existentes que pueden ser relevantes ya sea por que su fin es similar al nuestro o por las carencias que detectamos en ellas y que nos animan a desarrollar una solución mejor al mismo problema.

En este capítulo se expone la información obtenida al investigar acerca de los temas que se tratarán a lo largo del desarrollo del proyecto. Para ello, se expone en primer lugar un análisis de aplicaciones móviles y web relacionadas con los objetivos establecidos. A continuación se explica cuál es el entorno de desarrollo elegido y cuáles son sus características principales. Por último se explica la forma en la que se ha utilizado el desarrollo ágil como método para realizar el proyecto.

El estudio se va a realizar mediante la búsqueda a través de un buscador Web común como es el de Google y por medio del buscador de la aplicación Google Play disponible en todos los dispositivos Android. Los términos de búsqueda que se han utilizado en ambos medios han sido escritos tanto en castellano como en inglés y se exponen a continuación.

Las búsquedas en el buscador Web de Google han ido orientadas a aplicaciones o soluciones que permiten una localización en interiores más precisa que la ofrecida por redes móviles y GPS. Los términos de búsqueda utilizados para este caso se muestran a continuación:

- wifi indoor position maps
- indoor location solution

Por el contrario, en Google Play las búsquedas han sido relacionadas con aplicaciones móviles universitarias que hacen uso de mapas y geolocalización (al menos de lugares). Los términos de búsqueda definidos han sido los siguientes:

- universidad mapas
- university maps
- maps

Todas las búsquedas se realizaron durante la semana del 21 al 25 de noviembre de 2011 aunque las versiones analizadas en algunos casos tienen en cuenta las actualizaciones que han ido apareciendo desde entonces.

2.1 ANÁLISIS DE APLICACIONES MÓVILES Y WEB RELACIONADAS CON LOS OBJETIVOS ESTABLECIDOS

En primer lugar, para definir la investigación a realizar, se deben tener claros los principales objetivos de este PFC, que en este caso son los que se muestran a continuación:

- Crear una base de datos geolocalizada con las aulas, despachos y puntos de interés de los diferentes campus de la universidad.
- Crear una aplicación para el sistema operativo Android que permita la búsqueda de personas (PAS y PDI) y lugares utilizando la base de datos geolocalizada.

Teniendo en cuenta estos objetivos, se ha realizado una búsqueda partiendo del artículo de Alonso, García y Amescua [11] en el que se hace una investigación acerca del aprendizaje a través del móvil y de las aplicaciones móviles desarrolladas para las universidades españolas. En este artículo se concluye que solamente el 17,94% de las universidades españolas han entrado en el entorno de las aplicaciones móviles, además las aplicaciones que más fallan son las de plataforma Android probablemente debido a que al contrario que en el caso de iOS, las aplicaciones no son revisadas antes de publicarse. Por último, se concluye que por un lado, las aplicaciones encontradas son meramente informativas y no son demasiado complejas y por otro lado, no se explota demasiado la posibilidad de hacer aplicaciones que faciliten la vida universitaria.

Además de este artículo, se han realizado búsquedas en Google Play relacionando los términos universidad, mapa y geolocalización, de manera que las apps que se han encontrado son aplicaciones desarrolladas para la geolocalización en universidades de todo el mundo.

A continuación se describen las funcionalidades concretas que se han detectado en las aplicaciones más relevantes.

2.1.1 *GOOGLE MAPS*

La aplicación de mapas de Google para Android [1] ha sido la que nos ha servido de referencia durante todo el periodo de desarrollo del proyecto puesto que es la aplicación de mapas más utilizada en los dispositivos Android y por tanto a la que más usuarios están acostumbrados.

A finales de noviembre de 2011 apareció en la versión 6.0 una nueva funcionalidad que permite la visualización de planos de edificios sobre Google Maps. Para ampliar los conceptos expuestos, consultar [2].



Figura 2.1: Google Maps 6.0

2.1.2 POINT INSIDE SHOPPING & TRAVEL

La aplicación Point Inside [3] en su versión 3.1.0 permite buscar distintos tipos de servicios numerosos edificios alrededor del mundo. Por ejemplo el aeropuerto de Barajas. En estos edificios se pueden seleccionar las diferentes plantas y los diferentes servicios a buscar como pueden ser: restaurantes, tiendas, lavabos, etc.

Además, en el caso de lugares como un aeropuerto que se componen de varios edificios, permite elegir la planta a mostrar de un edificio en concreto para que el usuario preste atención únicamente al edificio en el que se está buscando.



Figura 2.2: Point Inside 3.1.0

2.1.3 INDOOR MAPS AND POSITIONING

La empresa Ericsson ha creado en este caso una API para desarrollo de aplicaciones móviles que utilizan posicionamiento en interiores [13]. Además de la API, tienen una aplicación denominada Map Studio que facilita la creación de los mapas.

La API facilita la unión de los mapas creados con Map Studio con la geolocalización basada en redes wifi.

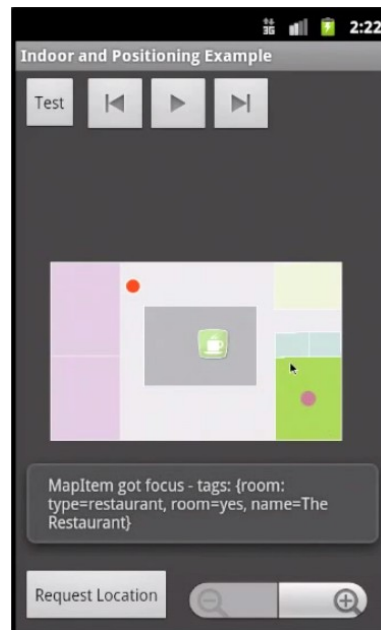


Figura 2.3: Aplicación de ejemplo utilizando Indoor Maps and Positioning

2.1.4 NAO CAMPUS

NAO Campus [12] es una solución de localización y navegación en interiores que permite al usuario ubicarse en el interior de edificios mediante la combinación de señales GPS y wifi.

Su característica más interesante es que permite definir mapas en tres dimensiones y crear una base de datos georreferenciada de manera que las aplicaciones que las utilicen estarán provistas de una funcionalidad muy interesante que es el poder saber a qué altura se encuentra un usuario, es decir, en qué planta del edificio.

Esto era un problema con las tecnologías GPS puesto que los satélites están a una altura desde la cual no se puede diferenciar con mucha precisión a qué altura se encuentra el receptor. Los errores actuales están en torno a los 20 metros. Por tanto en un edificio de 3 plantas cuya altura máxima son unos 20 metros no tiene sentido utilizarlo, puesto que puedes estar en la planta baja y por el error el GPS indica que estás en la tercera.

2.1.5 UNIVERSIDAD DE MURCIA

La aplicación corporativa de la universidad de Murcia [4] contiene un módulo de geolocalización de edificios, pero no permite buscar aulas o despachos.



Figura 2.4: Univ. Murcia 2.3

2.1.6 UNIVERSITY OF ALABAMA

La aplicación corporativa de la universidad de Alabama [5], al igual que la de Murcia, contiene un módulo de geolocalización de edificios. Permite buscarlos por su nombre, pero no permite buscar las aulas ni los despachos que contiene cada uno.

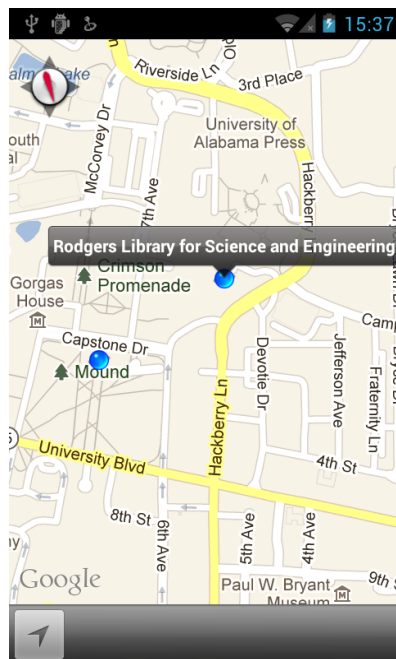


Figura 2.5: Alabama 1.5.2

2.1.7 BALL STATE UNIVERSITY MAP APP

La aplicación de mapas de la Ball State University [6] de Muncie, Indiana, permite la búsqueda de edificios, departamentos y lugares de interés.

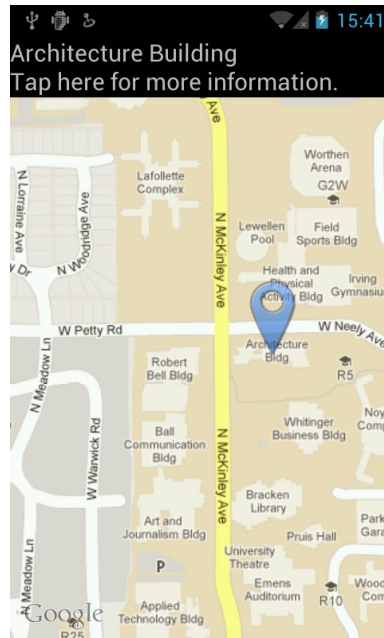


Figura 2.6: Ball State Map Application 1.0

2.1.8 MIT MOBILE

La aplicación corporativa del Massachusetts Institute of Technology (MIT) [7] contiene en su versión 2.3.1 un apartado de búsqueda de edificios, departamentos y servicios similar a la de Ball State University, por lo cual no se pueden buscar personas ni despachos o aulas. Los únicos datos georreferenciados son los centros de cada edificio.

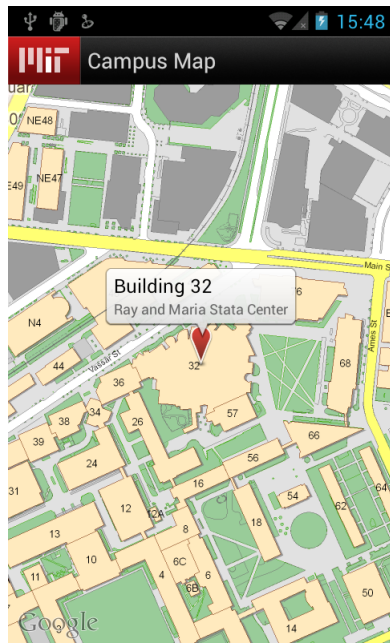


Figura 2.7: MIT Mobile 2.3.1 (Vista de mapa)



Figura 2.8: MIT Mobile 2.3.1 (Vista de detalle)

Después de explicar las características de cada una de las aplicaciones mencionadas, se realiza un análisis comparativo entre ellas teniendo en cuenta algunos puntos como son:

- La utilización de la API de Google Maps
- La georreferenciación de edificios
- La georreferenciación de lugares característicos dentro de los edificios
- La posibilidad de buscar lugares mediante un buscador
- La posibilidad de geoposicionar al usuario en altura
- La posibilidad de ver un plano del interior del edificio superpuesto en el mapa

El resultado de la comparación se puede consultar en la Tabla 2.1 que se muestra a continuación.

	Google Maps	Edificios	Lugares de interés	Búsqueda	Geoposición en altura	Plano superpuesto
Google Maps	Sí	No	Sí	Sí	No	Sí
Point Inside	Sí	No	Sí	Sí	No	Sí
Indoor maps & positioning	N. A.	N. A.	N. A.	N. A.	Sí	Sí
NAO Campus	N. A.	N. A.	N. A.	N. A.	Sí	N. A.
U. Murcia	Sí	Sí	No	No	No	No
U. Alabama	Sí	Sí	No	Sí	No	No
U. Ball State	Sí	Sí	Sí	Sí	No	No
MIT	Sí	Sí	No	Sí	No	No

Tabla 2.1: Comparativa entre aplicaciones

2.2 ENTORNO DE DESARROLLO MÓVIL SELECCIONADO

En esta sección se analiza en primer lugar el estado de los diferentes entornos móviles, a continuación se da una visión general sobre el sistema Android y se explica la estructura interna del sistema. Por último se comenta a nivel general cuál es el entorno de desarrollo integrado (EDI) utilizado y el porqué de la elección.

2.2.1 ENTORNOS MÓVILES

Los entornos móviles más comunes en España durante este último año y por orden de utilización son Symbian, Android, iOS, BlackBerry y Windows Phone.

Tal y como se muestra en la Figura 2.9, el uso de Android está en torno al 24% [10]. El hecho de que Symbian sea el primero puede llamar la atención, pero hay que tener en cuenta que hay muchos teléfonos móviles corrientes que utilizan este sistema operativo, es decir, no se consideran Smartphones.

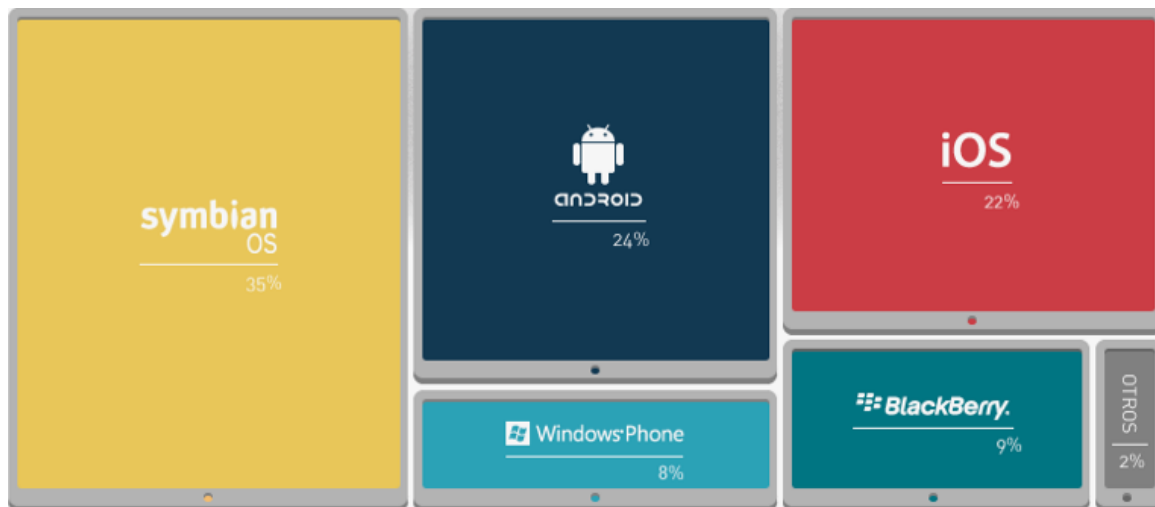


Figura 2.9: División del mercado de S.O. móviles¹

Por tanto, teniendo en cuenta estos datos, se decide elegir Android como la plataforma sobre la que desarrollar la aplicación móvil.

2.2.2 VISIÓN GENERAL SOBRE ANDROID

El entorno de desarrollo móvil elegido para la realización de este proyecto ha sido el sistema operativo Android debido a que cada vez su número de usuarios aumenta más en España y se cree conveniente llegar a la mayor parte de usuarios posibles.

Las versiones del sistema operativo Android, se nombran con un número y un nombre en clave que las define. En la Tabla 2.2 se listan las diferentes versiones junto con su nombre en clave desde la 1.5.

¹ Los datos han sido tomados en España en febrero de 2012

Nombre	Número de versión
“Cupcake”	Android 1.5
“Donut”	Android 1.6
“Eclair”	Android 2.1
“Froyo”	Android 2.2
“Gingerbread”	Android 2.3 - Android 2.3.2 Android 2.3.3 - Android 2.3.7
“Honeycomb”	Android 3.1 Android 3.2
“Ice Cream Sandwich”	Android 4.0 - Android 4.0.2 Android 4.0.3 - Android 4.0.4

Tabla 2.2: Nomenclatura de las versiones de Android

La versión 1.0 del sistema operativo Android se lanzó el día 21 de octubre de 2008. Android está basado en el sistema Linux y fue concebido como un sistema operativo para dispositivos móviles como son los Smartphone (teléfonos inteligentes) y las tabletas (tabletas inteligentes).

El desarrollo de Android está llevado a cabo por la Open Handset Alliance, cuyo liderazgo depende de la empresa Google.

La versión actual es la 4.0.4 “Ice Cream Sandwich” la cual ha incorporado una gran cantidad de cambios con respecto a las anteriores ya que unifica las versiones para Smartphone y tableta. Pero debemos tener en cuenta que actualmente, la fragmentación en Android es muy grande debido al elevado número de versiones que tiene este sistema y a que de todas ellas existen usuarios.

Este hecho puede comprobarse observando la Figura 2.10, en la que se muestra la distribución de versiones instaladas en los dispositivos Android de todo el mundo.

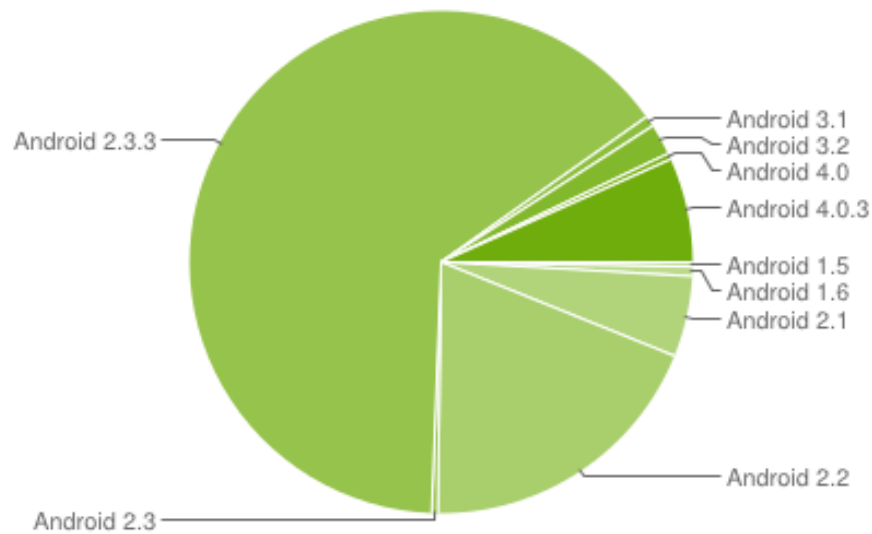


Figura 2.10: Versiones de Android funcionando actualmente²

Como se puede observar en la Figura 2.10, la versión más extendida actualmente es “Gingerbread”, instalada aproximadamente en el 65% de los terminales Android.

2.2.3 ESTRUCTURAL DEL SISTEMA ANDROID

Tal como se describe en el post redactado por el usuario *plusminus* en el foro de anddev.org [17], el sistema Android tiene una serie de componentes que se muestran en la Figura 2.11 y se explican a continuación.

² Los datos han sido tomados a nivel mundial en junio de 2012



Figura 2.11: Arquitectura del sistema Android

Aplicaciones

Android viene con una serie de aplicaciones básicas incluyendo un cliente de correo electrónico, una aplicación de SMS, un calendario, una aplicación de mapas, un navegador Web, una aplicación de contactos y algunas más. Todas estas apps están escritas usando el lenguaje de programación Java.

Framework de aplicaciones

Los desarrolladores tienen total acceso a las mismas API utilizadas por las aplicaciones básicas. La arquitectura de aplicación está diseñada para simplificar la reutilización de todos los componentes; cualquier aplicación puede publicar sus capacidades para recibir “intenciones” y otras aplicaciones puedan hacer uso de esas capacidades (proporcionando restricciones de seguridad manejables cumplidas por el entorno de trabajo). Este mismo mecanismo permite que TODOS los componentes puedan ser reemplazados por el usuario.

Lo fundamental de todas las aplicaciones es un conjunto de servicios y sistemas incluyendo:

Un rico y extensible conjunto de vistas que pueden ser utilizadas para construir una aplicación, incluyendo cuadrículas, listas, textViews, editTexts, selectores, botones y un navegador Web integrable e incluso un MapView que pueden ponerse en una aplicación con unas pocas líneas de código.

Contiene proveedores que permiten a las aplicaciones acceder a datos de otras aplicaciones (como la de contactos), o compartir sus propios datos.

Un gestor de recursos automático, haciendo que los recursos que nos son código (gráficos, cadenas de caracteres accesibles y archivos de disposición gráfica) sean accesibles desde el código.

Un gestor de notificaciones que permite a todas las aplicaciones mostrar alertas personalizadas en la barra de estado de la parte superior de la pantalla.

Un gestor de actividades que gestiona la vida de cada aplicación y permite una útil navegación hacia atrás.

Bibliotecas

Android incluye una serie de bibliotecas C/C++ utilizadas por varios componentes del sistema Android. Estas capacidades están expuestas a los desarrolladores a través del entorno de aplicaciones de Android. Algunas de las bibliotecas principales se listan bajo estas líneas:

- Biblioteca del sistema C – una implementación de la biblioteca del sistema estándar de C (libc), ajustada para dispositivos embebidos basados en Linux.
- Bibliotecas de medios – basada en OpenCORE de PacketVideo; las librerías soportan reproducción y grabación en muchos formatos de audio y vídeo populares, así como archivos estáticos de imagen, incluyendo MP3, AAC, AMR, MPEG4, H.264, JPG y PNG.
- SGL – el motor gráfico 2D fundamental.
- FreeType – representación de mapas de bits y fuentes vectoriales.
- Bibliotecas 3D – Una implementación basada en las APIs de OpenGL ES 1.0; las bibliotecas utilizan o aceleración hardware 3D (si está disponible) o las que lo incluya, un rasterizador software altamente optimizado para 3D.
- Gestor de Superficie – gestiona el acceso al subsistema de pantalla y sin problemas 2D y 3D compuestos de capas gráficas de múltiples aplicaciones.
- SQLite – un potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- LibWebCore – un moderno motor de navegador Web que impulsa tanto el navegador Android como el WebView.

Android en tiempo de ejecución

Android incluye una serie de bibliotecas del núcleo que proveen la mayoría de la funcionalidad disponible en las bibliotecas del núcleo del lenguaje de programación Java.

Cada aplicación Android se ejecuta en su propio proceso dado por el SO, y posee su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de manera que un dispositivo puede ejecutar múltiples máquinas virtuales eficientemente. La máquina virtual Dalvik ejecuta ficheros en formato *.dex* (Dalvik Executable) que fue optimizado para un uso mínimo de CPU y memoria. La máquina virtual está basada en registros y

ejecuta clases compiladas por un compilador de lenguaje Java que han sido transformadas en tiempo de compilación al formato *.dex* usando la herramienta “dx”, que está incluida en el SDK.

El Kernel de Linux puede ejecutar múltiples instancias de la máquina virtual Dalvik, también proporcionando funcionalidad como hilos y gestión de memoria a bajo nivel.

Kernel de Linux

Android depende de Linux (versión 2.6 del Kernel) para servicios del núcleo del sistema como gestión de memoria, gestión de procesos, pila de red, seguridad y modelo de controlador. El núcleo también actúa como una capa de abstracción del hardware entre las aplicaciones y el hardware.

2.2.4 ENTORNO DE DESARROLLO INTEGRADO

Tal y como se indica en el manual de instalación del SDK de Android [18] el IDE elegido para desarrollar la aplicación ha sido Eclipse Indigo en su última versión (la 3.7.2) puesto que ya he trabajado con él en muchas ocasiones a lo largo de la carrera y lo conozco suficientemente bien. Además, es el más utilizado por los desarrolladores de Android a nivel mundial y la cantidad de documentación es mayor.

Este entorno permite al desarrollador diseñar las interfaces gráficas con un asistente de gráfico sobre el que se pueden arrastrar los distintos componentes a las diferentes pantallas que aparecerán en la aplicación, de modo que en los ficheros Java no se tengan que especificar este tipo de detalles.

Eclipse permite también crear archivos XML mediante asistentes gráficos, de manera que el desarrollador no tenga que preocuparse por la sintaxis de los mismos sino por los valores que contienen.

También se integra mediante un complemento el ADB, de manera que al ejecutar el proyecto, en caso de que el desarrollador no disponga de un terminal con Android, se pueda elegir como dispositivo un dispositivo virtual que funcionará como un dispositivo Android estándar. Lo bueno es que se pueden probar las aplicaciones para diferentes versiones del sistema operativo y distintas resoluciones de pantalla, lo cual da al desarrollador la posibilidad de crear una aplicación que se pueda ejecutar en el mayor número de dispositivos posible.

Por último, la consola LogCat permite al desarrollador obtener información acerca de las ejecuciones mediante las trazas del emulador, los logs (registros) que decida el desarrollador y los errores o excepciones que se produzcan. De esta forma, es muy sencillo saber dónde se encuentran los errores a solucionar durante la fase de desarrollo.

2.3 DESARROLLO ÁGIL COMO MÉTODO PARA REALIZAR EL PROYECTO

Las Metodologías Ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales (ISO-9000, CMM, etc.) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

Las metodologías de desarrollo ágiles son adaptativas en lugar de predictivas. Los métodos de ingeniería tienden a intentar planear una parte grande del proceso del software en gran detalle para un plazo largo de tiempo, esto funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para estas metodologías, no obstante, el cambio es bienvenido.

Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos. Son orientadas a la gente y no orientados al proceso. Y su meta es definir un proceso que funcionará bien con cualquiera que lo use.

2.3.1 *THE AGILE ALLIANCE*

En febrero de 2001, tras una reunión celebrada en Utah (EE.UU.), nace el término ágil aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó “The Agile Alliance”, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía ágil, es este manifiesto se valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.
- La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

A continuación se muestran en forma de tabla las diferencias que existen entre las metodologías ágiles y las tradicionales:

Metodología Ágil	Metodología No Ágil (Tradicional)
Pocos artefactos	Más artefactos
Pocos roles	Más roles
No existe un contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes
Menos énfasis en la arquitectura	La arquitectura es esencial

Tabla 2.3: Metodologías Ágiles vs. Tradicionales

Dentro de las metodologías ágiles una de las características principales es el uso de historias de usuario. Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

2.3.2 SCRUM

Por último se va a hacer referencia a la metodología ágil utilizada como base para la realización de este proyecto que es Scrum [14] y cómo se ha adaptado a nuestro entorno.

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión (ROI) para su empresa. Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación.

Con Scrum, el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo le permite en cualquier momento realinear el software con los objetivos de negocio, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración.

Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades.

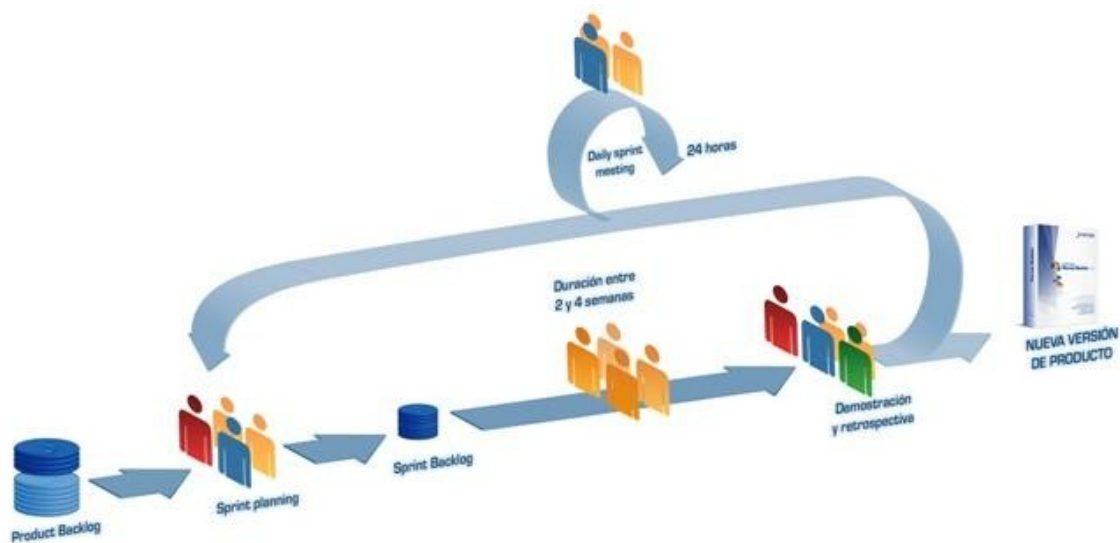


Figura 2.12: Diagrama de flujo de Scrum

Los principales beneficios que proporciona la utilización de Scrum son los siguientes:

- **Cumplimento de expectativas:** El cliente establece sus expectativas indicando el valor que le aporta cada requisito/historia del proyecto, el equipo los estima y con esta información establece su prioridad. De manera regular, en las demos de Sprint se comprueba que efectivamente los requisitos se han cumplido y se transmite feedback (retroalimentación) al equipo.
- **Flexibilidad a cambios:** Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado.
- **Reducción del Time to Market:** El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- **Mayor calidad del software:** La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
- **Mayor productividad:** Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
- **Maximiza el ROI:** Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.
- **Predicciones de tiempos:** Mediante esta metodología se conoce la velocidad media del equipo por sprint (los llamados puntos historia), con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está como trabajo pendiente.
- **Reducción de riesgos:** El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.

2.4 CONCLUSIONES DEL ANÁLISIS

En función de los resultados que hemos obtenido del análisis de otras aplicaciones, nos parece interesante la funcionalidad que ofrece la aplicación oficial de Google Maps, aunque se echa en falta la posibilidad de buscar edificios por su nombre, la cual aparece en la mayoría de aplicaciones universitarias estudiadas y la de determinar la posición del usuario en altura, que solamente está disponible en una de ellas en este momento.

En nuestro proyecto debemos incluir la funcionalidad que permite buscar a personas dentro de los edificios, la cual no existe en estas aplicaciones. Se mostrará el plano de la planta en la que se encuentra el lugar o persona buscado, de manera que el usuario sea capaz de orientarse en el interior del edificio. La geolocalización en altura parece muy útil, pero a su vez es compleja, por lo que en este proyecto no se va a tratar.

Para cubrir estas necesidades, se va a crear una aplicación móvil que permitirá la búsqueda de personas y lugares en los diferentes campus de la universidad, así como la consulta de información relevante acerca de los lugares y las personas, siendo posible comunicarse con las personas buscadas de manera sencilla.

En el desarrollo de esta aplicación móvil queremos utilizar una metodología de desarrollo ágil, puesto que en un principio, los requisitos de la aplicación no están definidos. En este caso, se elige Scrum que es la más adecuada a este tipo de desarrollos software, pero se ha adaptado para que las iteraciones sean de dos semanas. En esas iteraciones se parte de una reunión de los miembros del equipo entorno a la última versión del producto (si la hay) para definir las tareas a realizar durante la iteración por cada uno de los miembros del equipo de manera que al final exista una nueva versión del producto.

Durante el tiempo en que se llevan a cabo esas tareas, se hacen reuniones diarias para comprobar el avance del trabajo, para comentar cuestiones técnicas que no hubieran quedado claras en la reunión anterior o para plantear alternativas que se planteen en la siguiente reunión común.

CAPÍTULO 3 ESPECIFICACIÓN DE REQUISITOS

3.1 INTRODUCCIÓN	28
3.2 DESCRIPCIÓN GENERAL DEL SISTEMA A DESARROLLAR.....	30
3.2.1 PERSPECTIVA DEL PRODUCTO	30
3.2.2 FUNCIONES DEL PRODUCTO	30
3.2.3 CARACTERÍSTICAS DE LOS USUARIOS	30
3.2.4 INTERACCIÓN CON OTROS SISTEMAS	31
3.2.5 RESTRICCIONES.....	32
3.2.6 SUPOSICIONES Y DEPENDENCIAS	32
3.3 REQUISITOS ESPECÍFICOS	32
3.3.1 REQUISITOS FUNCIONALES	34
3.3.2 INTERFACES EXTERNAS	39
3.3.3 REQUISITOS DE RENDIMIENTO	44
3.3.4 RESTRICCIONES DE DISEÑO.....	44
3.3.5 ATRIBUTOS DEL SISTEMA.....	45

3.1 INTRODUCCIÓN

En este capítulo se realiza una Especificación de Requisitos Software (ERS). En ella se muestran los requisitos necesarios para el correcto desarrollo de la aplicación por parte del programador. Esta ERS está dirigida a todas las personas que deseen saber cuáles son los requisitos con los que cumple esta aplicación.

Para el correcto desarrollo de esta ERS se ha utilizado como guía el estándar IEEE 830 [8] que permite definir los requisitos mediante una serie de tablas muy sencillas de comprender.

El contexto en el que se concibe el sistema *UC3M Mapas* es el momento en el que la Universidad Carlos III de Madrid (UC3M) decide que desea desarrollar una serie de aplicaciones móviles destinadas a facilitar la vida universitaria proporcionando información y servicios útiles para las personas que estén relacionadas con la universidad de una forma u otra.

Cuando se toma esta decisión, se decide también que al pretender llegar al mayor número posible de usuarios, las aplicaciones se deben desarrollar en una primera versión para los sistemas iOS y Android.

Desde el primer momento, y basándose en aplicaciones móviles de universidades tanto españolas como internacionales, se decide crear una aplicación corporativa que sirva como entrada a las demás aplicaciones. Cabe destacar que las aplicaciones se pueden dividir en dos tipos: auto-contenidas e instalables.

Las aplicaciones auto-contenidas se consideran relevantes para todos los usuarios y por lo tanto, cuando se instala la aplicación corporativa, se instalan todas ellas en conjunto. En la siguiente lista se describe brevemente cada una de las aplicaciones de este tipo:

- **Noticias** ofrece un listado con las noticias de la UC3M. El usuario puede leerlas por separado y compartirlas a través de redes sociales.
- **Personas** ofrece la posibilidad de buscar sobre el directorio de la universidad los datos personales de cualquier miembro de la comunidad universitaria (Estudiante/PDI/PAS). Además, es posible agregar al contacto en la agenda del dispositivo y realizar acciones como llamadas telefónicas o envío de correos electrónicos a las personas encontradas.
- **Multimedia** ofrece la posibilidad de visualizar vídeos con información acerca de la actividad que se desarrolla dentro de la universidad. Ofrece la posibilidad de búsqueda y ordenación por distintos parámetros de los vídeos.
- **Estudios** ofrece un catálogo con la información de las titulaciones de Grado y Posgrado de manera que el usuario pueda consultar los Planes de Estudios de las diferentes titulaciones impartidas en la UC3M.
- **Cómo llegar** ofrece al usuario indicaciones para llegar a los distintos campus de la UC3M en función del medio de transporte seleccionado (coche, autobús, metro o tren).

Las aplicaciones instalables, además de ser más complejas que las auto-contenidas, se entiende que no tienen por qué estar instaladas en todos los dispositivos que tengan la aplicación corporativa instalada. Desde la pantalla principal de ésta, es posible instalar las aplicaciones instalables, de manera que al usuario le resulte muy sencillo añadir nueva funcionalidad a la aplicación.

En la siguiente lista se explican brevemente las aplicaciones instalables que se han decidido implementar como complemento de la corporativa:

- **Agenda** ofrece al usuario una lista con los eventos programados en los distintos campus de la UC3M clasificados por tipo. Los eventos se pueden consultar también a modo de calendario, de manera que al seleccionar un día, se ven los eventos programados ese mismo día. Ofrece también una vista de detalle de cada uno de los eventos que permite consultar la información al completo y compartir los eventos a través de redes sociales. Por último, ofrece una vista que ayuda al usuario a sacar el máximo partido de la aplicación explicando cada una de las funcionalidades de manera gráfica y sencilla.
- **Calificaciones** ofrece al estudiante consultar su expediente académico completo. Esta aplicación requiere inicio de sesión por parte de los usuarios, que además deberán ser estudiantes de la UC3M.
- **Mapas** ofrece una búsqueda geolocalizada de lugares y personas dentro de los campus de la UC3M. Muestra el plano del edificio en el que se encuentra la localización buscada. Desde la vista de detalle de lugar o persona se permite al usuario comunicarse con las personas buscadas y reportar errores por correo electrónico acerca de la geolocalización de los lugares. Existe la posibilidad de integrar la aplicación con otras aplicaciones ya que ofrece servicios de localización en la UC3M.

Esta última aplicación es la que se procede a desarrollar en este proyecto, junto con la base de datos georreferenciada y el repositorio de planos.

El nombre de la aplicación, cuando nos referimos a ella desde dentro de la aplicación corporativa es *Mapas*, pero si nos referimos a ella desde el exterior, su nombre es *UC3M Mapas*.

La aplicación *UC3M Mapas* es beneficiosa en cuanto a que al finalizar el desarrollo de la misma, existirá una base de datos georreferenciada de todos los campus de la universidad, la cual podrá ser utilizada para otros fines.

El principal objetivo de este sistema es desarrollar servicios que faciliten la geolocalización de lugares y personas dentro de los diferentes campus de la UC3M. Entre estos servicios, el primero en ser desarrollado será el de búsqueda.

Con este sistema se quiere conseguir que cualquier persona que se acerca a la universidad y sin conocer previamente los campus, sea capaz de encontrar cualquier lugar o persona que tenga un despacho en la universidad.

3.2 DESCRIPCIÓN GENERAL DEL SISTEMA A DESARROLLAR

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3.3, haciendo que sean más fáciles de entender.

3.2.1 PERSPECTIVA DEL PRODUCTO

La aplicación *UC3M Mapas*, además desarrollarse para iOS y Android, ha sido desarrollado en versión Web móvil, la cual permite a los usuarios que posean otros dispositivos acceder de manera muy similar a la información ofrecida por las aplicaciones nativas.

Como ya se ha dicho en la sección **¡Error! No se encuentra el origen de la referencia.**, el sistema a desarrollar ofrece una búsqueda geolocalizada de lugares y personas dentro de los campus de la UC3M, muestra el plano del edificio en el que se encuentra la localización buscada, desde la vista de detalle de lugar o persona se permite al usuario comunicarse con las personas buscadas y reportar errores por correo electrónico acerca de la geolocalización de los lugares y existe la posibilidad de integrar la aplicación con otras aplicaciones ya que ofrece servicios de localización en la UC3M.

El valor principal es la localización en interiores de lugares y personas de la UC3M. La localización se apoya en los planos de las plantas de los edificios de la universidad, mostrando de forma real dónde se encuentra el lugar o la ubicación de la persona que se busca dentro de la Universidad.

Los planos ayudan a encontrar el camino hacia el lugar buscado y se apoyan en códigos de colores para identificar puertas de tránsito común y escaleras.

3.2.2 FUNCIONES DEL PRODUCTO

Las principales funciones que la aplicación *UC3M Mapas* permitirá realizar se detallan a continuación:

- Buscar personas
- Buscar lugares
- Seleccionar el campus en el que se realizan las búsquedas
- Seleccionar el tipo de mapa a mostrar
- Mostrar vista de detalle de persona
- Mostrar vista de detalle de lugar
- Reporte de errores
- Mostrar vista de ayuda

3.2.3 CARACTERÍSTICAS DE LOS USUARIOS

Los usuarios de la aplicación serán usuarios de dispositivos con sistema operativo Android. Serán por un lado estudiantes de grado o posgrado y trabajadores (profesores, investigadores o personal asociado y de servicios) recién llegados a la universidad como visitantes esporádicos (asistentes a congresos, actuaciones en el auditorio, graduaciones,

reuniones, ferias de empleo, lecturas de trabajos de fin de grado, proyectos de fin de carrera, fin de máster o tesis doctorales).

También hay otros tipos de usuario como son los distintos servicios de la universidad interesados en georreferenciar su ubicación de manera que los demás usuarios puedan encontrarlos con facilidad.

El último tipo de usuario es un miembro la comunidad universitaria que se desplaza de su campus habitual de trabajo a otro campus que no conoce para realizar alguna actividad.

3.2.4 INTERACCIÓN CON OTROS SISTEMAS

La aplicación va a interactuar con varios sistemas externos cuyo mantenimiento no depende del desarrollador. Esto podrá generar problemas a los usuarios de la aplicación en caso de caída de servidores o modificaciones en el servicio ofrecido por los proveedores de servicio.

Los sistemas con los que interactúa la aplicación son los siguientes:

- El servicio de Google Maps que es el que ofrece la vista de mapa de la aplicación. Gracias a que este servicio está distribuido alrededor del mundo es muy poco probable que falle alguna vez y en ese caso el tiempo de recuperación será muy bajo.
- El servicio Web provisto por la universidad es el que permite buscar personas. En caso de que este servicio falle, la búsqueda de personas no funcionaría pero la de lugares no se vería afectada.
- El servicio Web provisto por el grupo de investigación *Software Engineering Lab (SEL-UC3M)*, que es el que se encarga de la búsqueda de lugares y es el que tiene acceso a la base de datos georreferenciada.
- Otras aplicaciones pueden hacer uso de los servicios de localización provistos por *UC3M Mapas* a través de su API de comunicación.
 - Por un lado, las aplicaciones Android pueden hacer uso de estos servicios. Como ejemplo tenemos la aplicación *UC3M Agenda* desde la cual se puede consultar el lugar en el que se celebra un evento, la aplicación *Personas* incluida en la aplicación corporativa de la UC3M desde la cual se puede consultar la posición del despacho de la persona buscada, o aplicaciones para congresos celebrados en los campus de la universidad como es la aplicación *ITGSM12* creada específicamente para el congreso ITGSM 2012 desde la cual se puede consultar a posición de las localizaciones utilizadas en el evento relacionadas con el horario de las ponencias.
 - Por otro lado, cualquier aplicación informática ejecutada en un dispositivo Android, puede hacer uso de la API ofrecida por *UC3M Mapas* a través de URLs específicas a las que se deben pasar unos parámetros concretos que permiten realizar búsquedas tanto de lugares como de personas.

El resto de comunicaciones se hacen dentro del dispositivo. Se llama desde la aplicación a otras aplicaciones como la de correo electrónico, las que permiten la opción de compartir (redes sociales, correo, mensajes y Bluetooth entre otras), la aplicación de contactos que permite agregar a personas a la agenda y la aplicación de teléfono que permite llamar a personas.

3.2.5 RESTRICCIONES

La interfaz gráfica de la aplicación deberá cumplir con las guías de diseño de Android [9]. La aplicación deberá ser compatible con dispositivos cuya versión de Android sea igual o superior a la 2.2 “Froyo”.

El sistema *UC3M Mapas* deberá disponer de una API/interfaz a través de la cual se podrá conectar desde otras aplicaciones. La interfaz recibirá parámetros para realizar búsquedas.

Los lenguajes de programación no están restringidos por el cliente, pero se utilizará JAVA por ser el más indicado para desarrollar aplicaciones Android.

La comunicación con el servidor se realizará a través de un servicio Web que devuelve información en formato JSON y que la aplicación procesará para mostrar los resultados de las consultas.

Los planos que se guarden en el repositorio deberán cumplir los requisitos establecidos en el OpenGIS Web Map Tile Service Implementation Standard [15].

La criticidad del sistema reside en el tiempo de respuesta de las consultas.

Es necesario que el sistema sea seguro ante ataques como XSS, etc.

3.2.6 SUPOSICIONES Y DEPENDENCIAS

El sistema *UC3M Mapas* hace uso de un servicio Web que accede a través de otro servicio Web ofrecido por el Servicio de Informática (SdI) al directorio de la universidad. En caso de que uno de los dos servicios Web dejara de funcionar por alguna razón, la aplicación no funcionaría en modo búsqueda de personas.

La búsqueda de lugares se vería comprometida si el primer servicio Web no funcionara, puesto que la función que realiza no depende del servicio ofrecido por el SdI.

3.3 REQUISITOS ESPECÍFICOS

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas.

En la sección 3.3.1 se muestran los requisitos funcionales que son los que tratan sobre la funcionalidad que proporciona la aplicación.

En la sección 3.3.2 se muestran las interfaces externas. Por interfaces externas se entienden tanto las vistas de la aplicación con las que puede interactuar directamente el

usuario. También se consideran interfaces externas las que ofrecen a otras aplicaciones conectarse con *UC3M Mapas* y las interfaces que conectan la aplicación con servicios externos.

La sección 3.3.3 es la que contiene los requisitos que afectan al rendimiento de la aplicación.

En la sección 3.3.4 se exponen las restricciones de diseño que limitan ciertos puntos en la aplicación.

Para definir los requisitos se utiliza una tabla con una serie de campos cuyos posibles valores se muestran en la siguiente tabla de ejemplo.

ID	Identificador		Nombre	Requisito de ejemplo	
Prioridad	Alta	X	Necesidad	Esencial	X
	Normal	X		Deseable	X
	Baja	X		Baja	X
Dependencias	Otro Identificador		Fecha	dd/mm/aaaa	
Descripción	Descripción del requisito. Explicación que permita a cualquier programador, implementarlo si necesidad de más información.				

Tabla 3.1: Identificador - Requisito de ejemplo

Los identificadores elegidos para los requisitos constan de dos partes bien claras separadas por un guión. La primera es un código de dos letras que identifica el tipo de requisito y la segunda es un número de tres cifras que sirve para ordenar los requisitos entre los de su mismo tipo y comenzando por el 001.

Los códigos de dos letras que identifican el tipo se explican a continuación:

- RF – Requisito funcional
- RI – Requisito de interfaz externa
- RR – Requisito de rendimiento
- RD – Restricción de diseño

3.3.1 REQUISITOS FUNCIONALES

ID	RF-001		Nombre	Selección del tipo de búsqueda	
Prioridad	Alta	X	Necesidad	Esencial	X
	Normal			Deseable	
	Baja			Baja	
Dependencias			Fecha	10/10/2011	
Descripción	Debe ser posible seleccionar el tipo de búsqueda (lugar o persona) de manera que dependiendo de la opción seleccionada se realice un tipo u otro de búsqueda.				

Tabla 3.2: RF-001 - Selección del tipo de búsqueda

ID	RF-002		Nombre	Buscar según un término de búsqueda	
Prioridad	Alta		Necesidad	Esencial	X
	Normal	X		Deseable	
	Baja			Baja	
Dependencias	RF-001		Fecha	10/10/2011	
Descripción	<p>El sistema debe mostrar en el mapa la ubicación correspondiente al término de búsqueda indicado teniendo en cuenta el campus y el tipo de búsqueda seleccionados por configuración.</p> <p>En caso de búsqueda de persona:</p> <ul style="list-style-type: none">• Se mostrará la ubicación del despacho que tiene asignado.• Si no tiene despacho asignado, se indicará que la persona buscada no se ha encontrado en ese campus. <p>En caso de búsqueda de lugar:</p> <ul style="list-style-type: none">• Se mostrará la ubicación del mismo.• Si no se encuentra, se indicará que el término de búsqueda no se ha encontrado en ese campus.				

Tabla 3.3: RF-002 - Buscar según un término de búsqueda

ID	RF-003		Nombre	Mostrar detalles del lugar	
Prioridad	Alta		Necesidad	Esencial	X
	Normal	X		Deseable	
	Baja			Baja	
Dependencias	RF-002		Fecha	2/11/2011	
Descripción	El sistema debe permitir consultar los detalles del lugar que se ha obtenido como resultado tras una búsqueda de lugar. Los datos que se mostrarán son los siguientes: código, despacho, edificio y campus.				

Tabla 3.4: RF-003 - Mostrar detalles del lugar

ID	RF-004		Nombre	Mostrar detalles de la persona	
	Prioridad	Alta	Necesidad	Esencial	X
		Normal		Deseable	
		Baja		Baja	
Dependencias	RF-002		Fecha	2/11/2011	
Descripción	El sistema debe permitir consultar los detalles de la persona que se ha obtenido como resultado tras una búsqueda de persona. Los datos que se mostrarán son los siguientes: nombre y apellidos, departamento/unidad, correo electrónico, edificio, campus y teléfono.				

Tabla 3.5: RF-004 - Mostrar detalles de la persona

ID	RF-005		Nombre	Reportar incidencia		
	Prioridad	Alta	X	Necesidad	Esencial	
		Normal			Deseable	X
		Baja			Baja	
Dependencias		RF-003, RF-004		Fecha	21/11/2011	
Descripción		Desde cualquiera de las vistas de detalle (lugar y persona) se deberá reportar una incidencia al administrador de la aplicación vía la aplicación predeterminada de correo electrónico de Android. El correo estará predefinido con un destinatario, un asunto y un esquema de texto que se deberá utilizar como guía para reportar el error correctamente.				

Tabla 3.6: RF-005 - Reportar incidencia

ID	RF-006		Nombre	Enviar correo electrónico a la persona	
Prioridad	Alta	X	Necesidad	Esencial	
	Normal			Deseable	X
	Baja			Baja	
Dependencias	RF-004		Fecha	14/11/2011	
Descripción	Desde la vista de detalle de persona será posible abrir la aplicación predeterminada de correo de Android para enviar un mensaje a la persona buscada.				

Tabla 3.7: RF-006 - Enviar correo electrónico a la persona

ID	RF-007		Nombre	Llamar a la persona	
Prioridad	Alta	X	Necesidad	Esencial	
	Normal			Deseable	X
	Baja			Baja	
Dependencias	RF-004		Fecha	14/11/2011	
Descripción	Desde la vista de detalle de persona será posible abrir la aplicación de teléfono de Android para llamar a la persona buscada.				

Tabla 3.8: RF-007 - Llamar a la persona

ID	RF-008		Nombre	Agregar a la persona como contacto	
Prioridad	Alta	X	Necesidad	Esencial	
	Normal			Deseable	X
	Baja			Baja	
Dependencias	RF-004		Fecha	14/11/2011	
Descripción	Desde la vista de detalle de persona será posible agregar a la persona como contacto en la agenda de contactos del terminal.				

Tabla 3.9: RF-008 - Agregar a la persona como contacto

ID	RF-009		Nombre	Mostrar el Twitter del SEL	
Prioridad	Alta		Necesidad	Esencial	
	Normal			Deseable	X
	Baja	X		Baja	
Dependencias	RI-005		Fecha	20/03/2012	
Descripción	El sistema permitirá abrir la aplicación de Twitter o la página Web a través del navegador para consultar el perfil de SEL-UC3M en esta red social, de manera que sea posible que el usuario pueda hacerse seguidor de la cuenta con facilidad.				

Tabla 3.10: RF-009 - Mostrar el Twitter del SEL

ID	RF-010		Nombre	Compartir la aplicación en redes sociales	
Prioridad	Alta		Necesidad	Esencial	
	Normal			Deseable	X
	Baja	X		Baja	
Dependencias	RI-005		Fecha	20/03/2012	
Descripción	Desde la vista de <i>acerca de</i> será posible compartir un texto predefinido a través de redes sociales, de manera que otros usuarios puedan conocer la aplicación.				

Tabla 3.11: RF-010 - Compartir la aplicación en redes sociales

ID	RF-011		Nombre	Mostrar los créditos	
Prioridad	Alta		Necesidad	Esencial	
	Normal	X		Deseable	X
	Baja			Baja	
Dependencias	RI-005, RI-006		Fecha	20/03/2012	
Descripción	Desde la vista de <i>acerca de</i> será posible abrir la vista de créditos en la que se muestra una imagen con un e-mail de contacto y los nombres de los implicados en la realización del proyecto.				

Tabla 3.12: RF-011 - Mostrar los créditos

ID	RF-012		Nombre	Seleccionar campus	
Prioridad	Alta		Necesidad	Esencial	X
	Normal	X		Deseable	
	Baja			Baja	
Dependencias	RI-004		Fecha	17/10/2011	
Descripción	El sistema permitirá desde la vista de configuración, seleccionar el campus en el que se desean realizar las búsquedas (Colmenarejo, Getafe o Leganés).				

Tabla 3.13: RF-012 - Seleccionar campus

ID	RF-013		Nombre	Seleccionar tipo de plano	
Prioridad	Alta		Necesidad	Esencial	
	Normal	X		Deseable	X
	Baja			Baja	
Dependencias	RI-004		Fecha	17/10/2011	
Descripción	El sistema permitirá desde la vista de configuración, seleccionar el tipo de plano que se desea mostrar (normal o satélite).				

Tabla 3.14: RF-013 - Seleccionar tipo de plano

ID	RF-014		Nombre	Traducción al inglés	
Prioridad	Alta	X	Necesidad	Esencial	
	Normal			Deseable	X
	Baja			Baja	
Dependencias			Fecha	12/12/2011	
Descripción	Todos los textos de la aplicación deben estar traducidos al inglés, de modo que un usuario que tenga su dispositivo configurado en inglés, vea todos los textos disponibles de la aplicación en este idioma, y no en castellano, que es el idioma original.				

Tabla 3.15: RF-014 - Traducción al inglés

3.3.2 INTERFACES EXTERNAS

En esta sección se describen los requisitos que afectan a la interfaz de usuario, a la interfaz con otras aplicaciones Android y la interfaz de comunicación a través de URL.

ID	RI-001		Nombre	API de comunicación	
Prioridad	Alta		Necesidad	Esencial	
	Normal	X		Deseable	X
	Baja			Baja	
Dependencias			Fecha	16/01/2012	
Descripción	El sistema será capaz de recibir parámetros de otras aplicaciones Android mediante un envoltorio (bundle). Estos parámetros serán: <i>function</i> , <i>idCampus</i> y <i>term</i> , todos ellos de tipo cadena de caracteres.				

Tabla 3.16: RI-001 - API de comunicación

ID	RI-002		Nombre	Comunicación a través de URL	
Prioridad	Alta		Necesidad	Esencial	
	Normal	X		Deseable	X
	Baja			Baja	
Dependencias			Fecha	21/02/2012	
Descripción	El sistema será capaz de recibir parámetros desde una URL con un formato concreto. Los parámetros serán: <i>function</i> , <i>idCampus</i> y <i>term</i> , todos ellos de tipo cadena de caracteres. El formato de la URL será el siguiente: http://apps.sel.inf.uc3m.es/apps/maps/				

Tabla 3.17: RI-002 - Comunicación a través de URL

ID	RI-003		Nombre	Vista de búsqueda	
Prioridad	Alta	X	Necesidad	Esencial	X
	Normal			Deseable	
	Baja			Baja	
Dependencias			Fecha	10/10/2011	
Descripción	El sistema mostrará una vista con un campo de texto, un botón y un mapa. Además, sobre el mapa habrá un texto que indicará el campus en el que se está buscando y, en caso de que se muestre algún plano, el edificio y la planta que se está mostrando.				

Tabla 3.18: RI-003 - Vista de búsqueda

ID	RI-004		Nombre	Vista de configuración	
Prioridad	Alta	X	Necesidad	Esencial	X
	Normal			Deseable	
	Baja			Baja	
Dependencias			Fecha	24/10/2011	
Descripción	El sistema mostrará una vista con que permita seleccionar el campus en el que se desea buscar (Leganés, Getafe o Colmenarejo) y el tipo de plano a mostrar (Satélite o Normal).				

Tabla 3.19: RI-004 - Vista de configuración

ID	RI-005		Nombre	Vista de acerca de	
	Alta		Necesidad	Esencial	
	Normal	X		Deseable	X
	Baja			Baja	
Dependencias			Fecha	20/03/2012	
Descripción			El sistema mostrará una vista que muestre algunos datos acerca de la aplicación como son la versión, la dirección de correo electrónico de contacto, el usuario de Twitter del grupo de investigación que la ha desarrollado. También habrá opción de mostrar la vista de créditos y de compartir la aplicación a través de redes sociales.		

Tabla 3.20: RI-005 - Vista de acerca de

ID	RI-006		Nombre	Vista de créditos	
Prioridad	Alta		Necesidad	Esencial	
	Normal			Deseable	X
	Baja	X		Baja	
Dependencias			Fecha	20/03/2012	
Descripción	El sistema mostrará una vista en la que se muestren los créditos de la aplicación incluyendo el logotipo de la universidad, una dirección de correo electrónico corporativo y el nombre de los servicios involucrados en la creación de la aplicación.				

Tabla 3.21: RI-006 - Vista de créditos

ID	RI-007		Nombre	Vista de ayuda	
Prioridad	Alta		Necesidad	Esencial	
	Normal	X		Deseable	X
	Baja			Baja	
Dependencias			Fecha	16/01/2012	
Descripción	El sistema mostrará una vista que muestre un tutorial a base de capturas de pantalla comentadas. La navegación entre capturas se hará mediante dos botones en la parte inferior.				

Tabla 3.22: RI-008 - Vista de ayuda

ID	RI-008		Nombre	Comunicación con servicio Web	
Prioridad	Alta		Necesidad	Esencial	X
	Normal	X		Deseable	
	Baja			Baja	
Dependencias			Fecha	17/10/2011	
Descripción	La aplicación se conectará al servicio Web a través de una petición GET con los siguientes parámetros: <ul style="list-style-type: none">• <i>function</i> (find person2)• <i>idCampus</i> (1 → Leganés 2 → Getafe 3 → Colmenarejo)• <i>term</i> (el término que desee buscar el usuario)				

Tabla 3.23: RI-008 – Comunicación con el servicio Web

3.3.3 REQUISITOS DE RENDIMIENTO

ID	RR-001		Nombre	Tiempo máximo en las consultas	
Prioridad	Alta		Necesidad	Esencial	X
	Normal	X		Deseable	
	Baja			Baja	
Dependencias			Fecha	12/12/2011	
Descripción	Se establece que el tiempo máximo que puede tardar una consulta es de 5 segundos. En caso de que una consulta tarde más, se mostrará un error indicando que ha habido un problema con la conexión al servicio Web.				

Tabla 3.24: RR-001 - Tiempo máximo en las consultas

ID	RR-002		Nombre	Abrir navegador en redes públicas	
Prioridad	Alta		Necesidad	Esencial	
	Normal	X		Deseable	X
	Baja			Baja	
Dependencias			Fecha	28/11/2011	
Descripción	En caso de que el terminal esté conectado a una red wifi pública se abrirá el navegador para que el usuario pueda iniciar sesión.				

Tabla 3.25: RR-002 - Abrir navegador en redes públicas

3.3.4 RESTRICCIONES DE DISEÑO

La única restricción de diseño impuesta por el cliente ha sido la relacionada con la versión del sistema operativo, que en un inicio fue la versión 2.3 y posteriormente se modificó a la versión 2.2 debido a que se detectó que era más interesante llegar a más usuarios.

ID	RD-001		Nombre	Versión del Sistema Operativo	
Prioridad	Alta		Necesidad	Esencial	X
	Normal	X		Deseable	
	Baja			Baja	
Dependencias			Fecha	3/10/2011	
Descripción	La aplicación deberá ser compatible con dispositivos a partir de la versión 2.2 “Froyo” del sistema operativo Android.				

Tabla 3.26: RD-001 - Versión del Sistema Operativo

ID	RD-002		Nombre	Versión del SDK de Java	
Prioridad	Alta		Necesidad	Esencial	X
	Normal	X		Deseable	
	Baja			Baja	
Dependencias			Fecha	17/10/2011	
Descripción	El lenguaje de programación a utilizar será JAVA y el nivel de compilación será el 1.6 debido a que Android 2.2 no acepta el 1.7 y a que la aplicación utiliza una biblioteca compilada en 1.6 y si el resto de la aplicación estuviera compilada en 1.5 habrá problemas.				

Tabla 3.27: RD-002 - Versión del SDK de Java

3.3.5 ATRIBUTOS DEL SISTEMA

La aplicación deberá ser fiable, es decir, deberá tener en cuenta todos los posibles errores “controlables” desde ella misma y aportará al usuario información relativa a estos errores de manera que el usuario pueda intentar solucionarlos si de él dependen.

La aplicación deberá poder ser mantenida sin mucha dificultad por parte de otro desarrollador utilizando la documentación disponible.

La aplicación no deberá ser portable ya que en este caso es específica para Android. Por el contrario, sí será compatible con el mayor número de versiones.

La aplicación deberá estar preparada para responder frente a ataques. En este caso, la capa de seguridad integrada en la parte del servidor, de manera que en la aplicación móvil no se tratan aspectos de seguridad.

CAPÍTULO 4 DISEÑO DE LA APLICACIÓN

4.1 PAPER PROTOTYPE	48
4.2 ALTERNATIVAS DE DISEÑO	58
4.2.1 ALTERNATIVA DE DISEÑO 1.....	62
4.2.2 ALTERNATIVA DE DISEÑO 2.....	63
4.2.3 ALTERNATIVA DE DISEÑO 3.....	63
4.3 DIAGRAMA DE CLASES.....	64
4.4 DIAGRAMAS DE SECUENCIA	69
4.5 DISEÑO DE LA BASE DE DATOS.....	75
4.6 DISEÑO DEL REPOSITORIO DE PLANOS	76

En este capítulo se presentan en primer lugar, los *paper prototypes* (prototipos en papel) modificados en cada una de las iteraciones en las que había cambios que afectaran de manera considerable a la interfaz de usuario. Estos *paper prototypes* son muy útiles para plasmar gráficamente los requisitos que se tienen para una aplicación, dándole a conocer al cliente un vistazo de cómo va a lucir la aplicación y pulir detalles antes de que se pase a una siguiente etapa de diseño o desarrollo.

A continuación se explican las alternativas de diseño que se han barajado incluyendo sus ventajas e inconvenientes. Al final de esta sección se explica cuál ha sido la alternativa elegida y el porqué de la elección.

Posteriormente se presenta el diagrama de componentes para que el lector pueda disponer de una vista panorámica del diseño realizado.

En el siguiente apartado se muestran los diagramas de clases (uno por cada componente) de manera que se visualicen las relaciones entre las clases que involucran el sistema.

Una vez realizados los diagramas de clases se presentan los diagramas de secuencia necesarios para que un programador sea capaz de implementar la interacción de los objetos del sistema modelada en estos diagramas.

A continuación se explica brevemente y por medio de un diagrama Entidad/Relación y un relacional, el diseño realizado antes de crear la base de datos de lugares georreferenciados con la que interactuará la aplicación a través del servicio Web.

Por último, se explica el diseño elegido para la estructura que deberá tener el repositorio de planos y el proceso a realizar desde los planos originales hasta los distintos directorios que se colocarán dentro de este repositorio.

4.1 PAPER PROTOTYPE

En este apartado se muestra y explica en primer lugar el *paper prototype* (prototipo en papel) realizado en el mes de octubre de 2011, al inicio del proyecto

En la Figura 4.1 se muestra un boceto de lo que será la vista principal de la aplicación, en la que se muestra un botón de configuración en la parte superior derecha de la pantalla, una barra de búsqueda acompañada de un botón para que el usuario sea capaz de realizar las búsquedas y por último una vista de mapa en la que se dispondrán los resultados a las consultas hechas por el usuario.

Al pulsar sobre el botón de configuración se abrirá la vista de configuración mostrada en la Figura 4.2, en la que el usuario podrá seleccionar sus preferencias de tipo de mapa a mostrar, campus en el que realizar las búsquedas y tipo de búsqueda a realizar.

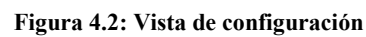




Figura 4.3: Vista de lugar



Figura 4.4: Vista ampliada

A lo largo del desarrollo de la aplicación se ha ido modificando el diseño de la aplicación y a continuación se muestra y explica el *paper prototype* final, creado en el mes de abril de 2012.

En la Figura 4.5 podemos ver cómo debería ser la pantalla principal de la aplicación. Contiene una cabecera con el título, un campo de texto y un botón para realizar las búsquedas, el mapa en el que se muestran los resultados y un selector en la parte baja que permite cambiar el tipo de búsqueda, dependiendo de lo que se quiera buscar (persona o lugar).



Figura 4.5: Pantalla principal

En la Figura 4.6 se muestra cómo se introduce el término de búsqueda en el campo de texto, para posteriormente pulsar el botón y realizar la búsqueda.

Antes de pulsar el botón hay que cerciorarse del tipo de búsqueda que se va a hacer para no obtener errores no deseados.



Figura 4.6: Introducción de término de búsqueda

En la Figura 4.7 se muestra el plano de la planta del edificio donde se encuentra el resultado de búsqueda. Además, se muestra un pop-up en el que se indica el código del lugar, el edificio y la planta.

En el caso de que el lugar buscado no esté al aire libre, existe la opción de mostrar los detalles del lugar pulsando en el botón que se encuentra en el pop-up.



Figura 4.7: Resultado de búsqueda de lugar

En la Figura 4.8 se muestran los detalles de ese lugar en concreto.



Figura 4.8: Detalles de lugar

En la Figura 4.9 se muestra, al igual que en la Figura 4.7 un resultado de búsqueda, pero en este caso la búsqueda es de persona, por lo cual, el texto de la barra de búsqueda y el del popup no tienen relación aparente.

Esto es porque el resultado mostrado es el despacho que la persona buscada tiene asignado.



Figura 4.9: Resultado de búsqueda de persona

Al igual que en la búsqueda de lugar, en la búsqueda de persona se pueden mostrar detalles, pero en este caso están relacionados con la persona y no tanto con el lugar.

En la Figura 4.10 se muestran los detalles de la persona buscada en la Figura 4.9.



Figura 4.10: Detalles de persona

Además, a través del botón de menú se accede a la configuración y a la vista de acerca de. En la Figura 4.11 se muestran las opciones desplegadas.



Figura 4.11: Opciones del menú principal

En la Figura 4.12 se muestran las opciones de configuración elegidas para cada categoría. Al pulsar en cada una de las categorías de la lista, se abre un diálogo en el que se puede seleccionar la opción deseada.

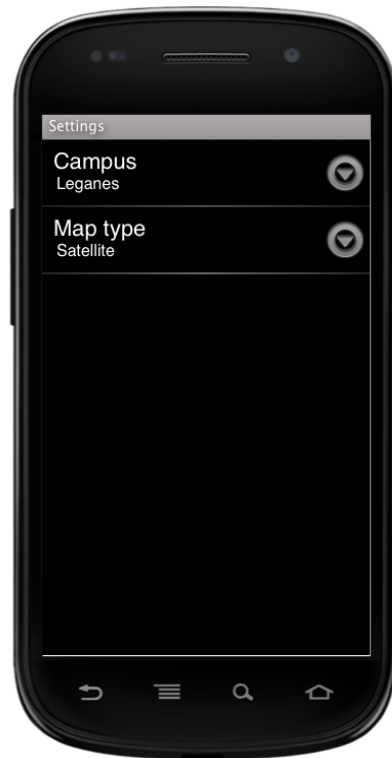


Figura 4.12: Pantalla de configuración

Por último, en la Figura 4.13 se muestra la vista con información acerca de la aplicación, en la que se exponen datos de contacto con los administradores, la versión de la y una forma sencilla de ver los créditos de la misma.

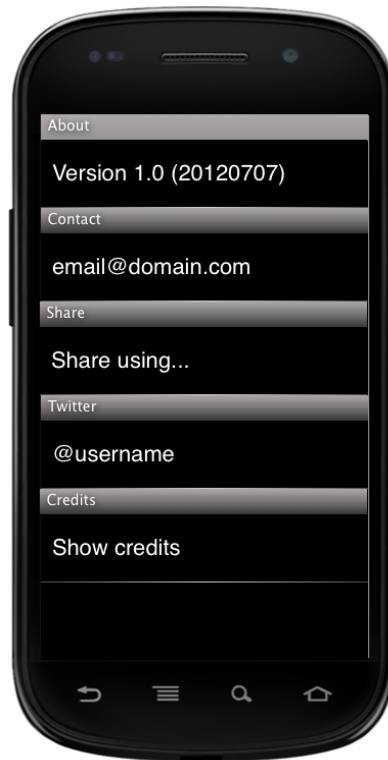


Figura 4.13: Vista de Acerca de

4.2 ALTERNATIVAS DE DISEÑO

En esta sección se explican brevemente las diferentes alternativas que se han barajado durante la fase de diseño de la aplicación y la que se ha elegido para el diseño definitivo.

Estas alternativas manejan una serie de opciones que pueden ser realizadas de formas diferentes y que se explican a continuación.

Servicio Web Al inicio del proyecto, tuvimos que decidir qué lenguaje íbamos a utilizar para devolver los datos que ofrecía el servicio Web. Las alternativas eran XML y JSON. La primera tiene como ventaja que es muy comprensible y existen muchas herramientas que permiten obtener en el cliente la información codificada. La opción de utilizar JSON tomó fuerza cuando nos dimos cuenta de que el servicio debía ser lo más rápido posible, y por lo tanto, cuanto menor fuera el número de bytes enviados, mayor sería la velocidad. El problema que tiene XML es que su estructura hace que haya que utilizar muchas etiquetas que lo único que hacen es añadir peso a los archivos. Por tanto, finalmente se decidió utilizar JSON.

Fuente de datos Cuando se diseñó el servicio Web, se pensó en recuperar la información de personas directamente del LDAP de la universidad. Esta opción finalmente se descartó puesto que para este fin es necesario utilizar un usuario y contraseña con permiso para acceder a dicho directorio, y se necesitaría una cuenta personal, lo cual no es muy recomendable. Por esta razón, se decidió comunicar nuestro

propio servicio Web con un servicio Web ofrecido por la universidad, de manera que nuestra aplicación no acceda al directorio directamente.

Longitud mínima del término de búsqueda La decisión de utilizar el servicio Web que nos proporcionaba la universidad nos causó algunos problemas puesto que tuvimos que ponernos en contacto con el administrador para definir la longitud mínima del término de búsqueda para que una búsqueda devolviera resultados ya que en caso de utilizar una letra como término de búsqueda puede hacer que los resultados sean de casi todo el directorio. Después de ver varios problemas se decidió que el número mínimo fuera tres, ya que uno de los campos por los que se puede buscar en el directorio es el identificador de usuario y hay algunos de esa longitud.

Formato de los datos devueltos Además, a nivel interno, tuvimos que definir cómo devolvía los datos nuestro servicio, puesto que hay personas con más de un despacho. Así que barajamos la posibilidad de devolver dos objetos de tipo persona con datos repetidos (excepto la localización), pero esto nos hacía recibir mucha información por duplicado, como es el correo electrónico, el nombre, los apellidos y el departamento en que trabaja.

Por todo esto, en el caso de la función *person2*, que es la que permite recuperar la información de las personas, decidimos que el formato fuera el siguiente:

```
[
  {
    "uid": 3840,
    "name": "Maria Isabel",
    "lastname": "Barro Vivero",
    "department": "Ingenieria Telematica",
    "email": "ibarro@di.uc3m.es",
    "locations": [
      {
        "roomNumber": "1.1.J05",
        "building": "Agustin De Betancourt",
        "campus": "Leganés",
        "telephone": "916249992"
      },
      {
        "roomNumber": "4.1.F15",
        "building": "Torres Quevedo",
        "campus": "Leganés",
        "telephone": "916249992"
      }
    ]
  }
]
```

Para la función *find*, que es la que devuelve la información de lugar, se decidió que el formato fuera el que se muestra a continuación:

```
[
  {
    "id_place": "240",
    "building": "Al Aire Libre",
    "buildingNumber": "0",
    "floor": "0",
    "lat": "40.332695",
    "lng": "-3.767110",
    "name": "Locomotora",
    "isRoom": "0"
  }
]
```

Autocompletado Otra de las alternativas de diseño que se pensaron durante el desarrollo fue la utilización de autocompletado en el campo de búsqueda.

Fijándonos en la aplicación de *Google Maps*[1], observamos que según el usuario escribe, la aplicación va ofreciendo distintas posibilidades, de manera que el usuario tiene facilidad a la hora de encontrar lo que busca sin necesidad de escribir el término de búsqueda completo.

El problema que encontramos cuando hicimos las primeras pruebas fue que cuando teníamos seleccionada la búsqueda de personas y se empieza a escribir, cuando se escribía la tercera letra, la aplicación hacía una consulta al servidor para obtener los resultados y poder mostrar las sugerencias. El problema era que con tres letras, el servicio Web devolvía 100 resultados (el tope máximo definido por el administrador del servicio) en la mayor parte de los casos y esto hacía que la aplicación se ralentizara excesivamente.

Así que se pensó en definir el número mínimo de caracteres para el autocompletado en 4 o 5. El problema seguía siendo parecido ya que si un usuario escribía mal una palabra y borraba para modificar el texto, cada vez que se borraba una letra, se volvía a hacer la consulta.

Definitivamente se decidió no implementar esta opción, ya que los problemas que surgían eran mayores que los beneficios que se iban a aportar con esta medida.

Fotografías de personas También en la fase de diseño se pensó que sería interesante poder ver, en la vista de detalle, la fotografía de la persona que se ha buscado, ya que por experiencia propia sé que hay un número considerable de miembros de la universidad a los que les cuesta menos recordar una cara que un nombre.

Esta característica se llegó a implementar pero se terminó descartando porque al no requerir la aplicación de inicio de sesión, se infringiría la Ley Orgánica de Protección de Datos (LOPD).

4.2.1 ALTERNATIVA DE DISEÑO 1

La primera alternativa de diseño consiste en que el servicio Web en formato JSON acceda directamente al LDAP de la universidad para obtener los datos personales de las personas buscadas a través de la aplicación.

En esta alternativa, es posible hacer uso del autocompletado, puesto que las consultas son relativamente rápidas cuando el número de caracteres utilizados para realizarlas es igual o mayor a cinco.

Los lugares se obtendrán de la base de datos georreferenciada con los lugares de los tres campus de la Universidad Carlos III de Madrid.

En la Figura 4.14 se puede ver un esquema del diseño planteado en esta alternativa.



Figura 4.14: Alternativa de diseño 1

La ventaja de esta alternativa es la velocidad, pero tiene dos inconvenientes:

1. Los datos del LDAP no están completamente sincronizados con las bases de datos personales de la universidad.
2. Los datos que devuelve el LDAP no tienen una estructura muy comprensible y por lo tanto es necesario realizar algunas acciones para devolverlos en un formato más manejable para la aplicación.

4.2.2 ALTERNATIVA DE DISEÑO 2

Esta segunda alternativa mostrada en la Figura 4.15 cambia la fuente de datos personales. En este caso, en lugar de utilizar el LDAP, se utiliza un servicio Web proporcionado por el Servicio de Informática (SdI) el cual devuelve datos actualizados y estructurados en formato JSON, al igual que el servicio Web de nuestro propio servidor, por tanto, las conversiones a realizar en estos datos son mucho más sencillas.

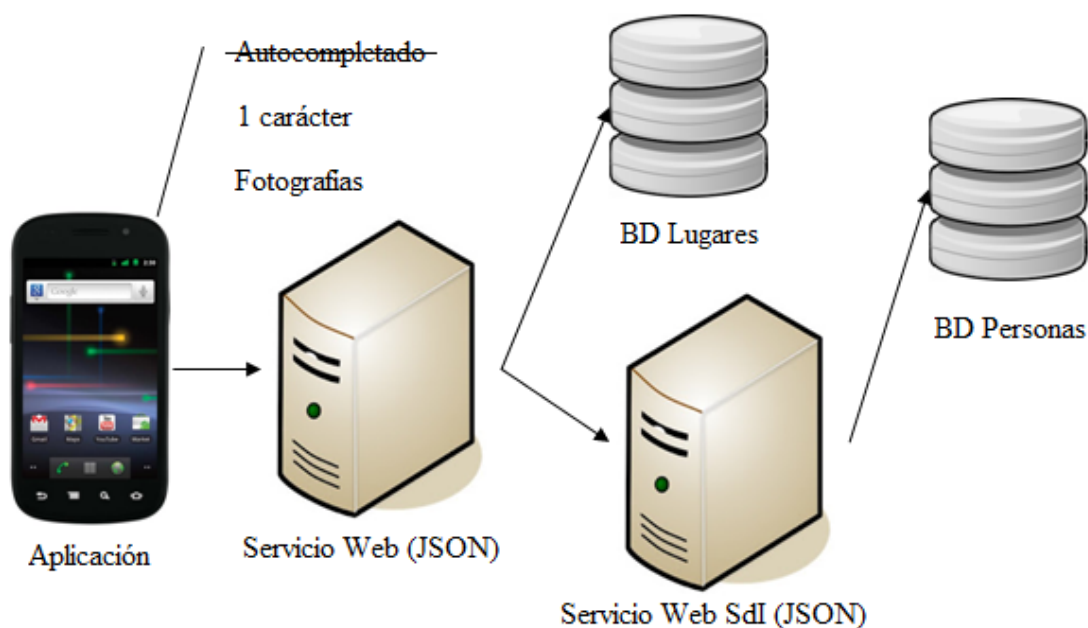


Figura 4.15: Alternativa de diseño 2

Este cambio hace que las consultas sobre datos de personas sean más lentas, por lo que se decide no utilizar el autocompletado, que ralentizaría la interacción del usuario de manera considerable.

Por lo tanto, al eliminar el autocompletado, se decide que las consultas en el caso de búsqueda de persona se puedan realizar incluso con un solo carácter.

Además, en esta alternativa se presenta la opción de incluir fotografías de las personas en la vista de detalle que proporciona la aplicación. Esta opción se presenta por el hecho de que el servicio Web proporcionado por el SdI devuelve entre otros datos, la URL de la fotografía de la persona.

4.2.3 ALTERNATIVA DE DISEÑO 3

En la Figura 4.16 se muestra la tercera alternativa que hemos barajado, que es igual que la anterior excepto por la opción que permite mostrar las fotografías de las personas en la vista de detalle.

La opción de las fotografías se ha descartado como se ha dicho antes por razones de legales de protección de datos personales.

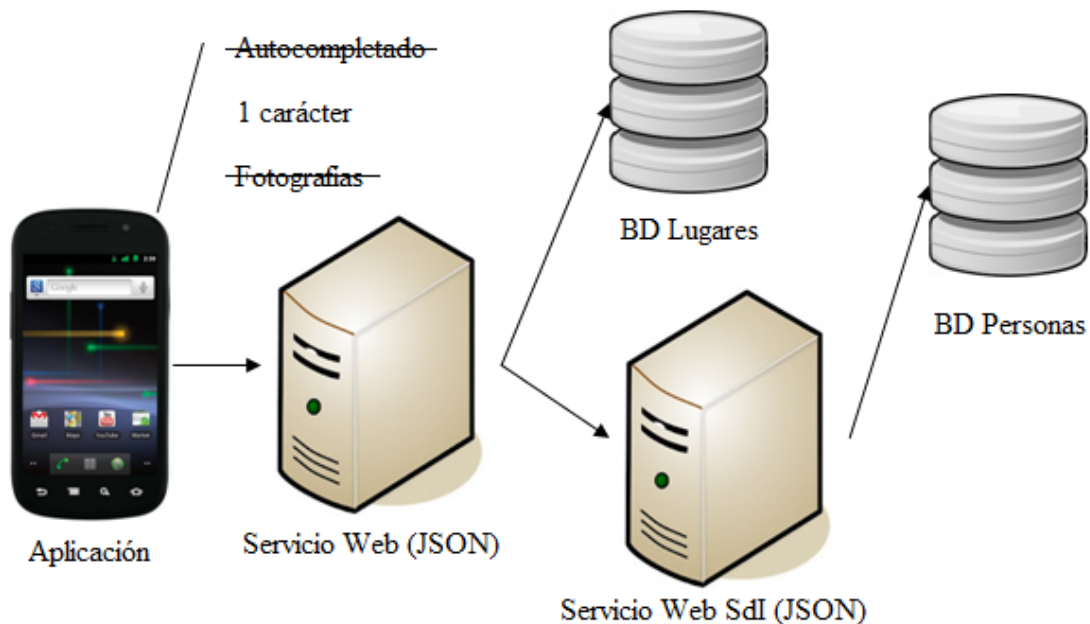


Figura 4.16: Alternativa de diseño 3

Esta última alternativa ha sido la elegida para el diseño final del sistema ya que es la más adecuada a nuestras necesidades.

4.3 DIAGRAMA DE CLASES

Después de haber elegido la alternativa a desarrollar, se ha realizado el diseño de la aplicación por medio de diagramas. Al ser el diagrama de clases muy extenso, se ha decidido mostrar en primer lugar un diagrama de componentes de manera que se comprenda la estructura general de la aplicación y posteriormente, varios diagramas de clases en los que se observe el contenido de cada uno de los componentes y las relaciones que existen entre las clases.

En la Figura 4.17 se muestra el diagrama de componentes en el que se pueden observar los siguientes componentes:

- **Server** es el servidor que consiste en una base de datos y un servicio Web que permite a la aplicación interactuar con ella.
- **osmdroid** es una biblioteca externa utilizada para poder mostrar los planos de los edificios superpuestos en el mapa.
- **Model** contiene clases que conforman el modelo de datos que se utilizarán en los demás componentes.
- **Lists** se encarga del funcionamiento de las listas que se utilizan para mostrar la información.
- **Controller** es un componente que posee clases con diferentes funcionalidades para ser utilizadas por el componente Mapas.

- **Mapas** es el componente principal de la aplicación y en el que se encuentra la mayor parte de la funcionalidad de la actividad principal de la aplicación.
- **Views** se encarga de controlar la forma en que se muestra el selector de búsqueda de persona o lugar.
- **Overlays** es el modulo que se encarga de define la forma en que se muestra la información que aparece sobre la vista de mapa.

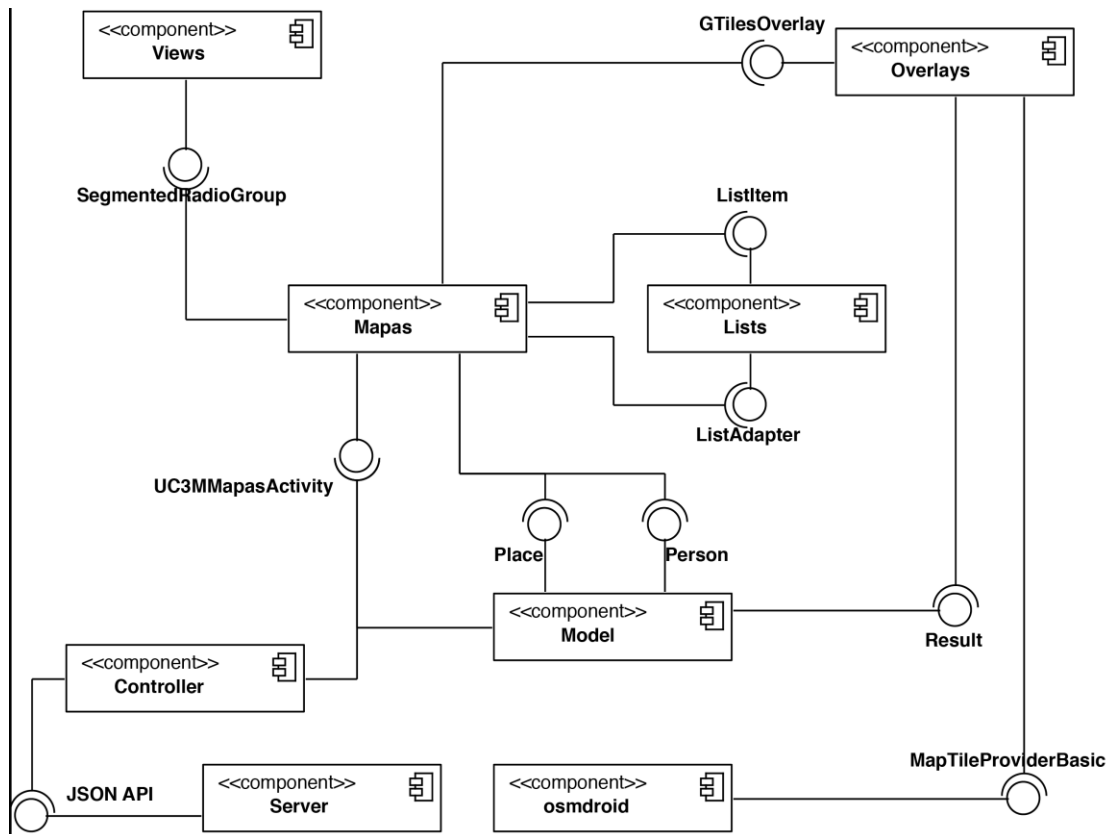


Figura 4.17: Diagrama de Componentes

Después de esta vista general del sistema se muestran los diagramas de clases de cada uno de los componentes anteriormente citados excepto los dos primeros (**Server** y **osmdroid**).

En la Figura 2.10 podemos encontrar una jerarquía de clases compuesta por la clase *Result* de la que heredan *Person* y *Place*. El sentido de esta jerarquía es que las búsquedas devuelvan un resultado de manera que desde la vista principal de la aplicación, al seleccionar el resultado (ya sea una persona o un lugar) se muestren de forma transparente los detalles.

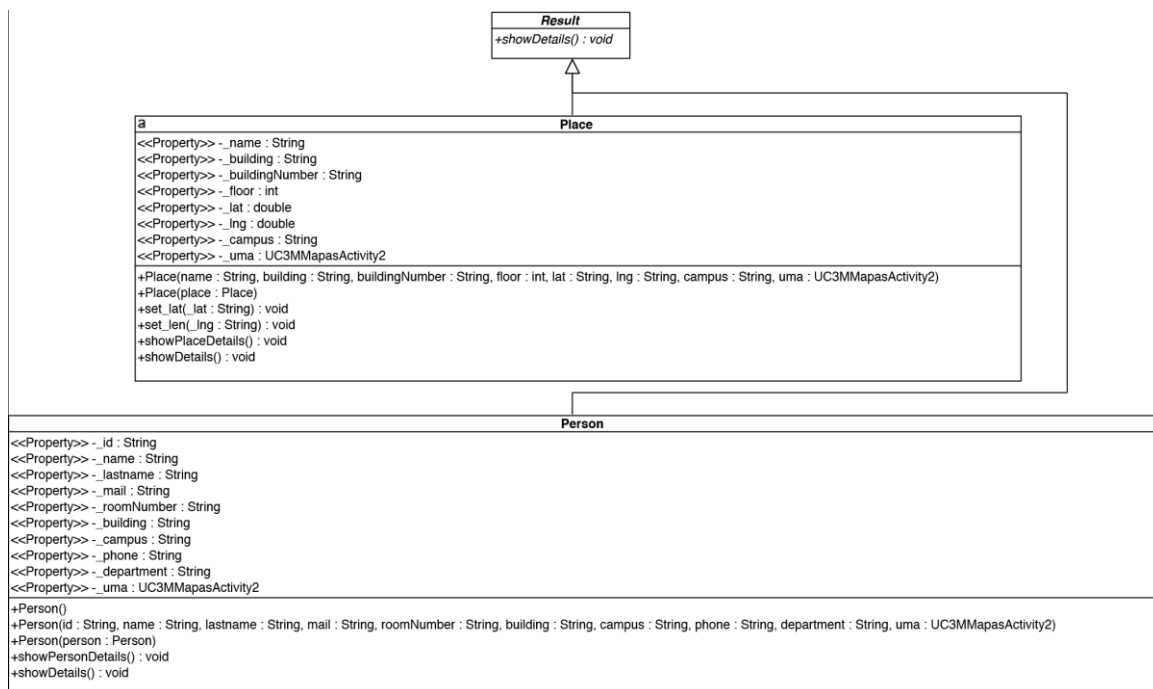


Figura 4.18: Model

En la Figura 4.19 se muestran dos clases. La clase *ListAdapter* recibe como parámetro en su constructor una lista de objetos de tipo *ListItem*, los cuales sirven para crear el adaptador que se pasa al *ListView* que se mostrará con las opciones deseadas.

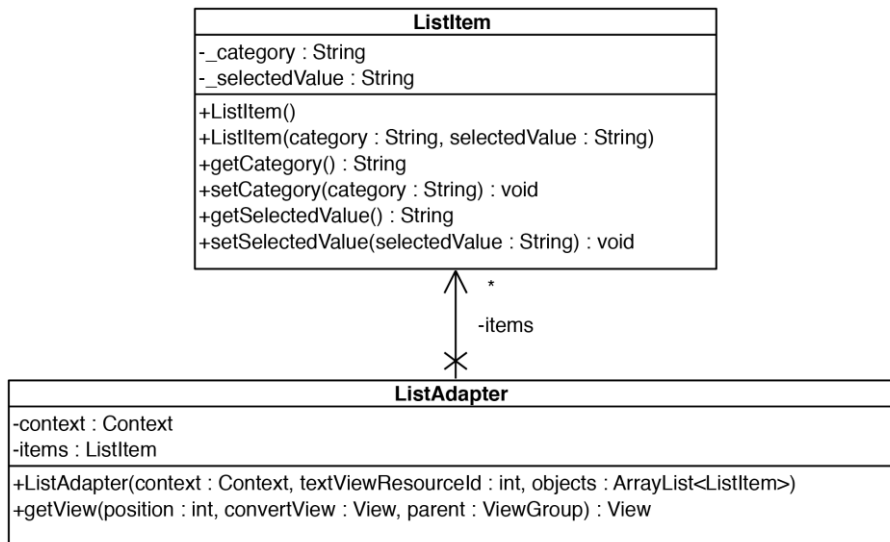


Figura 4.19: Lists

En la Figura 4.20 aparecen cinco clases independientes unas de otras, puesto que cada una tiene una funcionalidad diferente.

En primer lugar, la clase *Constants* no posee ninguna funcionalidad, sino que posee parámetros de configuración de la aplicación que puede cambiar el programador sin dificultad y que pueden ser utilizados por cualquier clase de la aplicación.

La clase *StringHandler* dispone de dos métodos que realizan unas operaciones algo complejas con cadenas de caracteres.

La clase *TestWifi* sirve para realizar una consulta HTTP de manera asíncrona y devolver un resultado. Es necesaria ya para que la aplicación sea compatible con la última versión de Android, la consulta debe hacerse fuera del hilo principal de la aplicación.

La clase *SearchTask* es una clase similar a la anterior pero en este caso, lo que devuelve es una matriz (array) de tipo JSON de manera que pueda ser analizado por los métodos de búsqueda que posee la aplicación en otras clases.

Por último, la clase *NetworkTester* posee métodos capaces de comprobar si la conexión a Internet existe o no, y en caso de que exista, si está funcional o no. En caso de estar en una red pública, se redirige al usuario al navegador para que pueda iniciar sesión directamente.

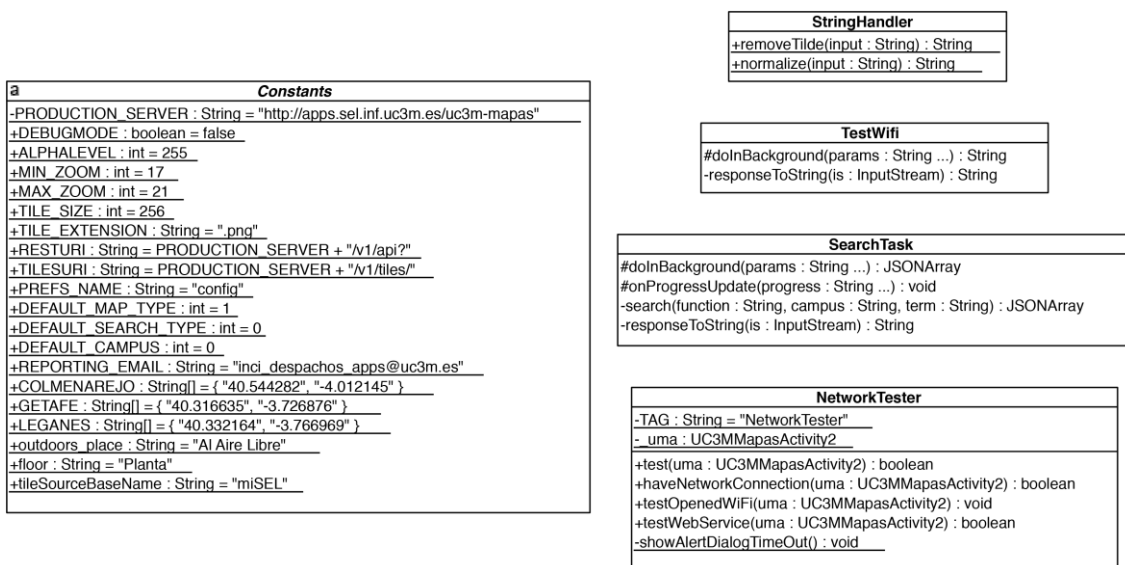


Figura 4.20: Controller

En la Figura 4.21 podemos ver el contenido del paquete principal cuya clase más importante es *UC3MMapasActivity*, la cual posee la mayor parte de la funcionalidad de la aplicación.

La clase *SettingsActivity* es la que controla la vista que permite al usuario modificar la configuración de la aplicación. Los parámetros a configurar son el tipo de mapa y el campus en el que se realizan las búsquedas.

La clase *PlaceActivity* es la que controla la vista que permite al usuario consultar los detalles de lugar. Gracias a ella se pueden reportar errores de geolocalización de lugares.

La clase *PersonActivity* es la que controla la vista que permite al usuario consultar los detalles de persona. Gracias a ella se puede enviar correos electrónicos a la persona buscada, se le puede agregar a la agenda como contacto, llamar por teléfono y reportar incidencias de geolocalización.

La clase *AboutActivity* es la que se encarga de la vista que muestra información acerca de la aplicación.

La clase *CreditsActivity* es la que se encarga de la vista que muestra los créditos de la aplicación.

La clase *HelpActivity* es la que se encarga de la vista de ayuda de manera que el usuario pueda aprender a utilizar la aplicación en todo su potencial de manera sencilla y rápida.

Por último, la clase *UC3MMapasActivity* es la que se encarga de la mayor parte de la funcionalidad de la aplicación. Se encarga de mostrar el plano correspondiente al campus seleccionado, superponer el plano del edificio concreto, recoger los parámetros de búsqueda para enviárselos a la clase que se encarga de las búsquedas, mostrar mensajes de error y manejar el menú principal.

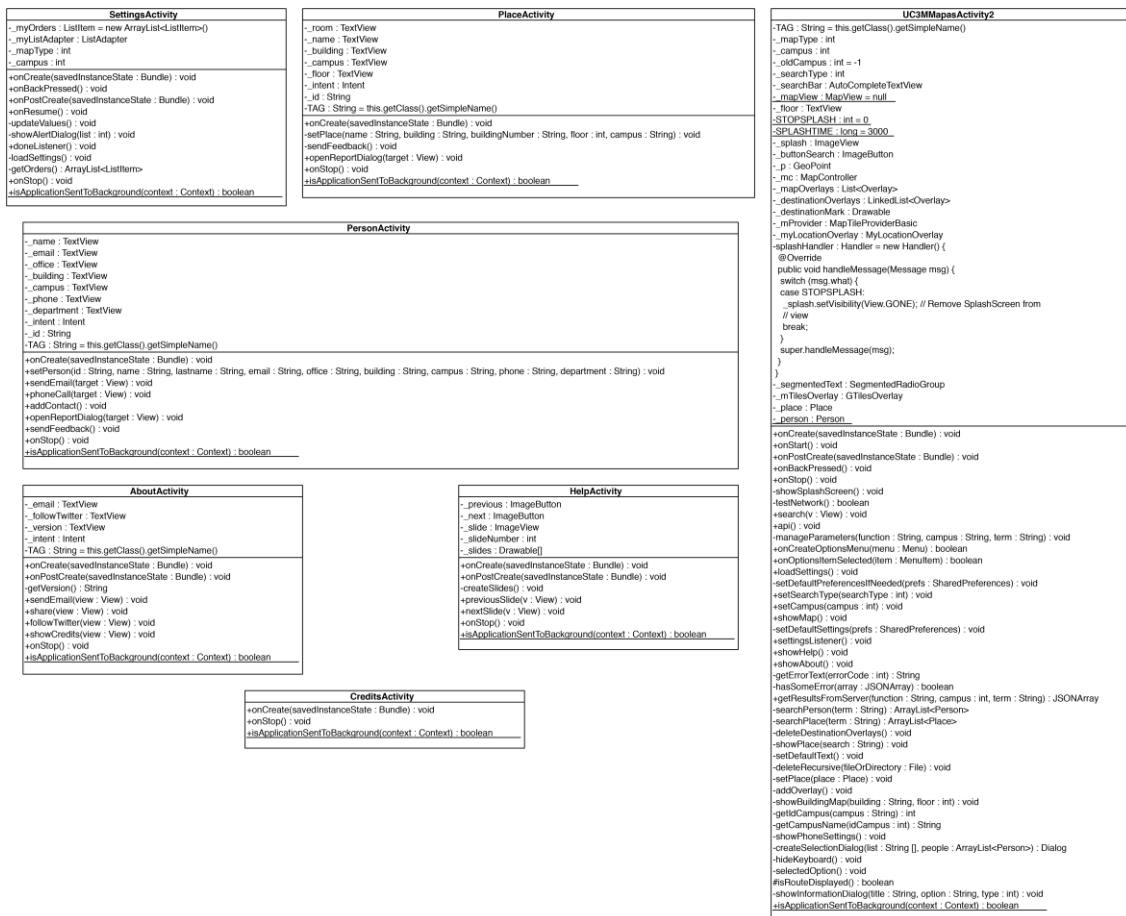


Figura 4.21: Mapas

En la Figura 4.22 se puede ver una única clase llamada *SegmentedRadioGroup* que es la que permite dibujar el botón segmentado que aparece en la parte baja de la pantalla principal de la aplicación.

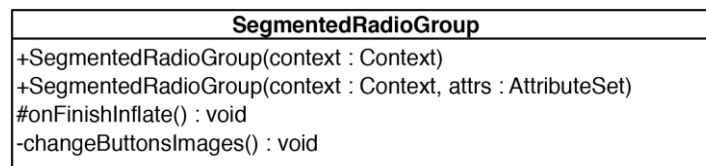


Figura 4.22: Views

En la Figura 4.23 se muestra el contenido del componente **Overlays**, en el que se encuentran algunas clases que permiten mostrar imágenes superpuestas en la vista de mapa.

La clase *MyItemizedOverlay* es la que se utiliza para crear los marcadores de posición de los resultados obtenidos tras las búsquedas. Esta clase hereda de la clase *BalloonItemizedOverlay*, la cual permite que cada marcador tenga asociada una burbuja en la que se muestra información cuando el usuario pulsa con su dedo sobre el marcador.

La clase *BalloonItemizedOverlay* posee un atributo de tipo *BalloonOverlayView* que es la que permite mostrar cierta información en la burbuja.

La clase *GTilesOverlay* que no posee relación con ninguna de las otras es la que nos permite definir algunos de los parámetros que se utilizan para mostrar los planos superpuestos de los edificios sobre la vista de mapa, como puede ser el nivel de transparencia que éstos tienen.

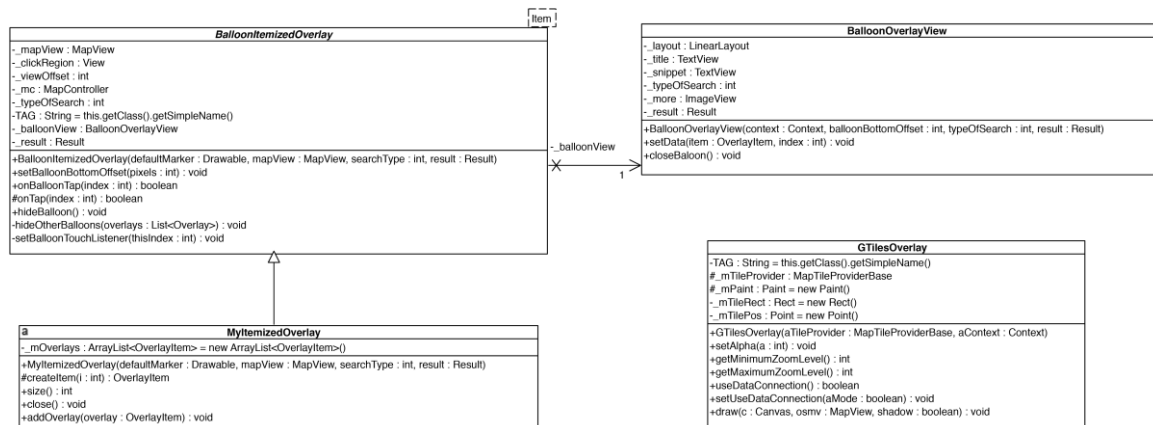


Figura 4.23: Overlays

Con esto quedan bien definidas las clases que se implementarán durante la fase de desarrollo de la aplicación y la separación en componentes que se utilizará para que la aplicación tenga más sentido.

4.4 DIAGRAMAS DE SECUENCIA

Una vez realizado el diagrama de clases en el que se muestran las relaciones entre las diferentes clases y componentes, se procede a definir los diagramas de secuencia que sirven al programador como modelo a la hora de desarrollar la interacción entre los objetos de la aplicación.

El primero de ellos se expone en la Figura 4.24 y muestra las acciones que se ejecutan al iniciar la aplicación *UC3M Mapas*.

En este diagrama se muestra un objeto del tipo *UC3MMapasActivity* sobre el cual se ejecutan una serie de llamadas que permiten que la aplicación se inicie correctamente. Estas acciones se listan ordenadamente a continuación:

1. Se muestra la pantalla de inicio que se muestra al abrir la aplicación mientras se carga la vista principal.

2. Se cargan las preferencias desde un archivo de configuración. En caso de que no exista el archivo, se creará con unos valores por defecto en ese mismo instante.

Se comprueba si el dispositivo tiene conexión a Internet.

3. En caso de que el dispositivo la tenga, se muestra el plano con los valores necesarios como son el punto central (el campus a mostrar), el nivel de zoom y el tipo de plano.
4. A continuación se ejecuta el método que comprueba si se han enviado parámetros a la aplicación y en cuyo caso, se procede a realizar las acciones necesarias para mostrar el resultado.
5. En caso de no haber conexión a Internet, se mostrará un diálogo informando al usuario del problema de conectividad e indicando que debe solucionarlo para poder iniciar la aplicación.

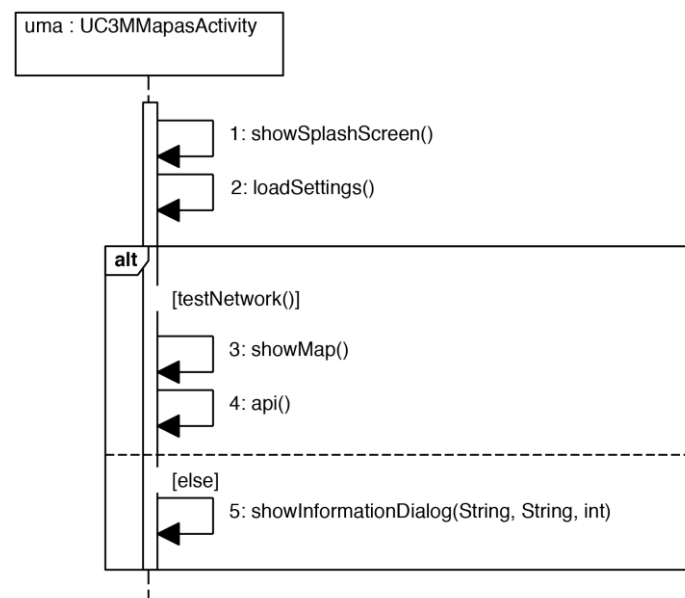


Figura 4.24: Inicio de la aplicación

En el siguiente diagrama mostrado en la Figura 4.25 se detalla la comprobación de la conexión del dispositivo a Internet que se realiza tanto al iniciar la aplicación como cada vez que se va a hacer una consulta al servicio Web, de manera que no haya errores de conexión inesperados. Bajo estas líneas se listan los pasos que se dan a la hora de comprobar la conexión:

1. Se ejecuta desde la clase *UC3MMapasActivity* el método estático de la clase *NetworkTester* denominado *test*. Este método realiza varias operaciones.

Se comprueba si el dispositivo está conectado o no a una red (wifi o 3G en la mayor parte de los casos).

- 1.1. En caso de estar conectado, se comprueba si la red es una red pública y en ese caso se abre el navegador para que el usuario inicie sesión antes de continuar. Si la red no es pública, se continúa la ejecución.

- 1.2. En caso de que la red no sea pública, se comprueba la conexión con el servicio Web, de manera que si funciona, el método finaliza y si no funciona, se muestra al usuario un diálogo informando del problema.
- 1.3. En caso de que el dispositivo no esté conectado a ninguna red, el método devuelve un valor de manera que se sepa que no hay conexión.

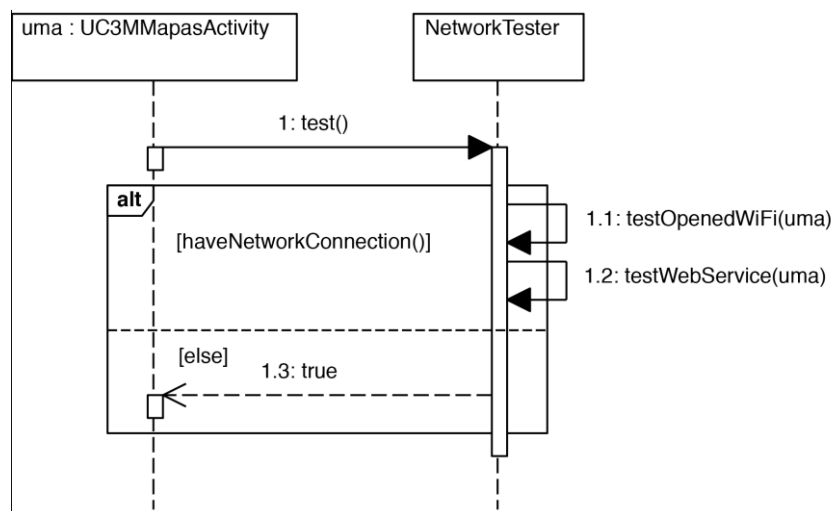


Figura 4.25: Comprobación de la conexión a Internet

El siguiente diagrama mostrado en la Figura 4.26 muestra la secuencia de pasos seguida cuando un usuario busca una persona a través de la aplicación.

A continuación se explican por orden los pasos a seguir en la ejecución de esta acción:

1. Como se ha explicado con el diagrama anterior, antes de nada, se comprueba si el dispositivo tiene conexión a Internet y al servicio Web.
2. En caso de haber conexión, se llama al método *search* de la clase *UC3MMapasActivity*.
3. A continuación se llama al método *showPlace*, el cual se encarga de recoger el valor introducido por el usuario en la barra de búsqueda.
4. En caso de que la barra de búsqueda no esté vacía, se lo pasa el valor al objeto *SearchTask* por medio de su método *doInBackground*.
 - 4.1. La clase *SearchTask* ejecuta su método *search* para conectarse al servidor y recibir la respuesta.
 - 4.2. Una vez recibida la respuesta, devuelve un objeto del tipo *JSONArray*.
5. La clase *UC3MMapasActivity* crea un objeto del tipo *Place* por cada uno de los lugares devueltos en el *JSONArray* resultante.
6. Cada uno de estos elementos se añade a la lista de lugares.
7. Se borran todos los resultados de búsquedas anteriores que pueda haber en la pantalla para no confundir al usuario.
8. Se muestran todos los lugares.

9. Por cada uno de los lugares guardados en la lista se muestra un marcador en el mapa.
10. Se borran del almacenamiento externo las imágenes que conforman los planos de los edificios, las cuales se descargan a medida que el usuario hace zoom en cuando se está mostrando un plano.
11. En caso de que en la lista haya más de un lugar, no se mostrará ningún plano. Pero en caso de que haya sólo un lugar, habrá que tener en cuenta qué edificio se quiere mostrar. Si el lugar no está en ningún edificio, tampoco se mostrará plano. Por el contrario, si el lugar está dentro de un edificio, se mostrará el plano correspondiente al edificio.

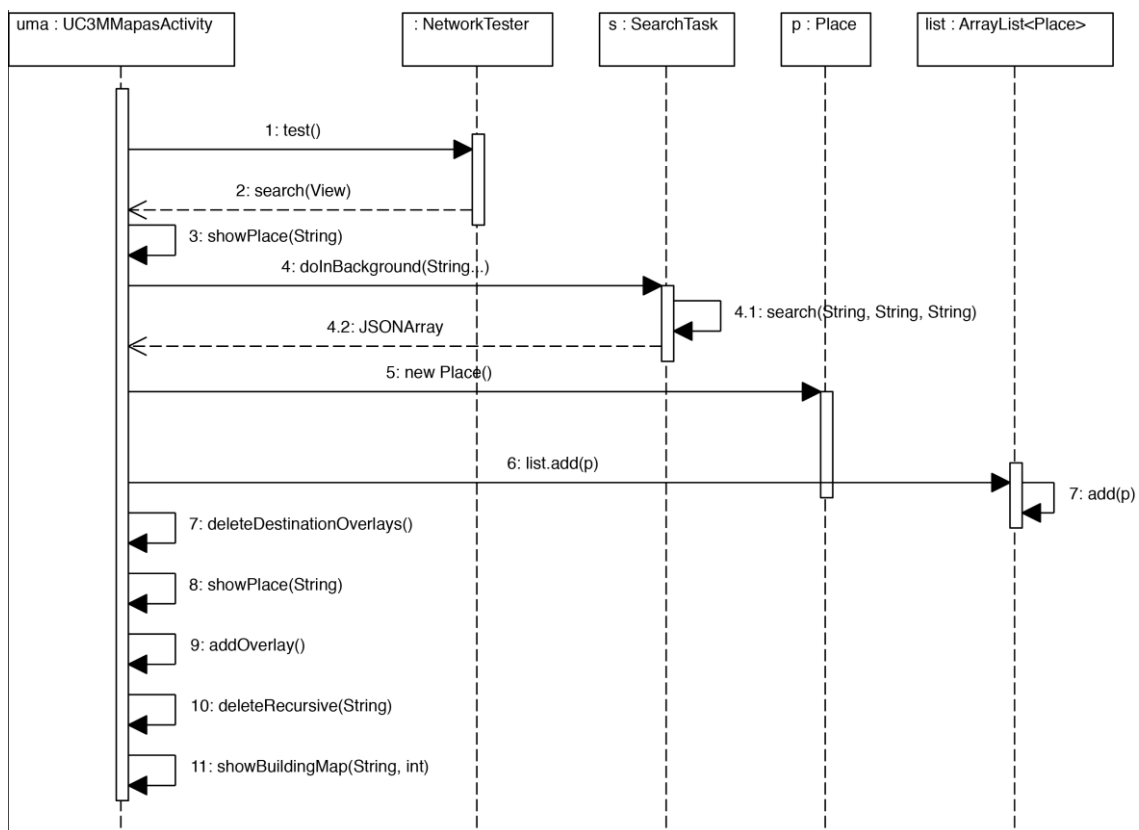


Figura 4.26: Búsqueda de un lugar

En el siguiente diagrama mostrado en la Figura 4.27 se muestra el proceso seguido cuando un usuario cambia la configuración de la aplicación.

Para ello, se siguen una serie de pasos explicados a continuación.

1. Gracias al botón de menú de la clase *UC3MMapasActivity*, se puede iniciar la clase *SettingsActivity* a través del método *startActivity*.
 - 1.1. En primer lugar, se cargan las preferencias desde el archivo generado en la primera ejecución de la aplicación.
 - 1.2. Se actualizan los valores de las etiquetas que se muestran en esta vista de manera que el usuario pueda ver cuál es la configuración actual.
 - 1.3. Cuando el usuario pulsa sobre una de las opciones

- 1.4. se muestra un diálogo con los posibles valores que puede tener el atributo. En caso de campus (Leganés, Getafe o Colmenarejo) y en caso de tipo de mapa (normal o satélite).
- 1.4.1. En el momento en que el usuario selecciona el valor deseado,
- 1.4.2. se actualizan los valores en memoria y
- 1.4.3. se cierra el diálogo.
- 1.5. Cuando el usuario pulsa el botón para volver a la pantalla principal
- 1.6. Las preferencias se guardan en el archivo de preferencias, de manera que la pantalla principal sea capaz de obtenerlos de ahí al volver a cargarse.
- 1.7. Finaliza la ejecución de la vista *SettingsActivity*.

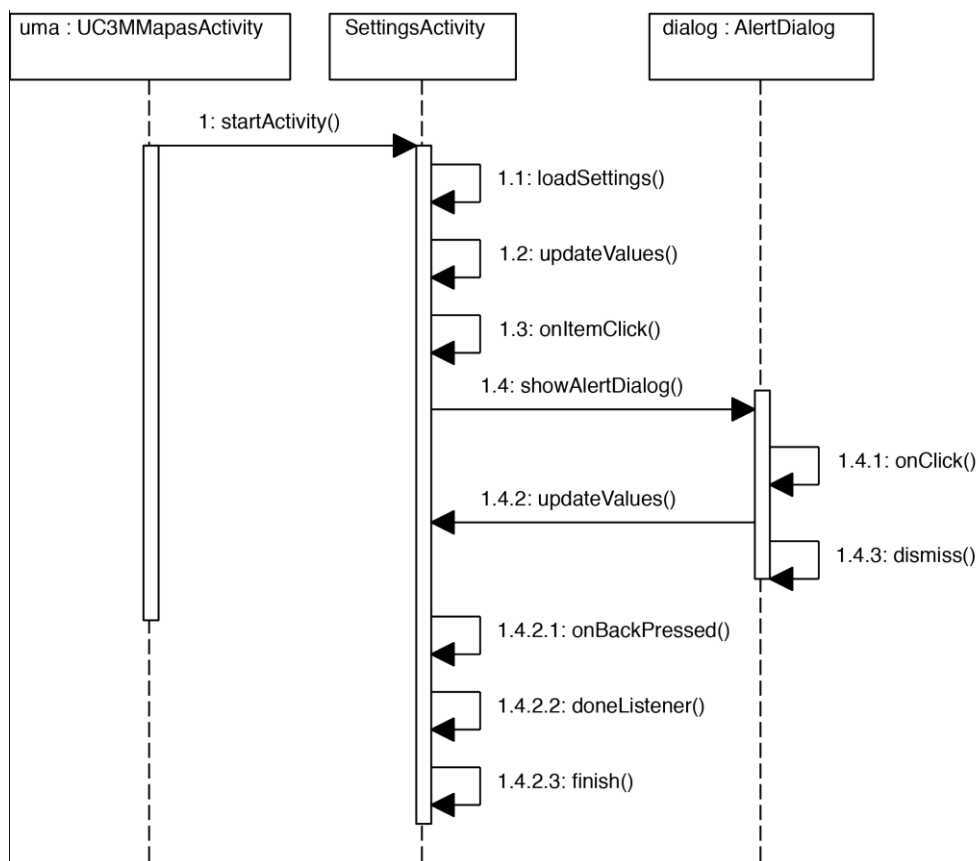


Figura 4.27: Modificación en la configuración

En este último diagrama de secuencia mostrado en la Figura 4.28 se muestran las acciones que se pueden realizar al mostrar la vista de detalle de persona.

A continuación se listan ordenadamente estas acciones:

1. En primer lugar, al botón que se muestra en la burbuja de información del marcador de posición se le añade un detector (*listener*) que permite saber cuándo el usuario pulsa sobre él.
2. El usuario pulsa sobre el botón.

- 2.1. Se llama al método *showDetails* del objeto *Result*, que es una abstracción del objeto *Person*.
 - 2.1.1. La clase *Person* llama a su método *showPersonDetails*.
 - 2.1.1.1. Este método inicializa el objeto de tipo *PersonActivity*.
 - 2.1.1.1.1. Se dan los valores a la persona de manera que se muestren en la pantalla.
3. El usuario pulsa sobre el correo electrónico mostrado y se ejecuta el método *sendEmail* de la clase *PersonActivity* el cual permite al usuario enviar un correo electrónico a través de la aplicación predefinida del sistema.
4. El usuario pulsa sobre el teléfono mostrado y se abre la aplicación de llamadas para que pueda llamar a la persona buscada.
5. El usuario pulsa sobre el botón de la parte superior de la pantalla que muestra un diálogo con una serie de acciones posibles. En el momento en que el diálogo se genera, se ejecutan dos métodos:
 - 5.1. Uno que crea un *listener* que en caso de que el usuario pulse, ejecutará el método que permite añadir a la persona a contactos y
 - 5.2. otro que crea un *listener* que en caso de que el usuario pulse, ejecutará el método que permite reportar un error.
6. El usuario pulsa la opción de añadir contacto, por lo que la aplicación de contactos se abre para que el usuario verifique los datos y acepte.
7. El usuario pulsa la opción de reportar un error, de manera que se abrirá la aplicación de correo con un texto predefinido que permite al usuario reportar el error sin mucho esfuerzo para que el administrador pueda recibir la información en un formato comprensible.

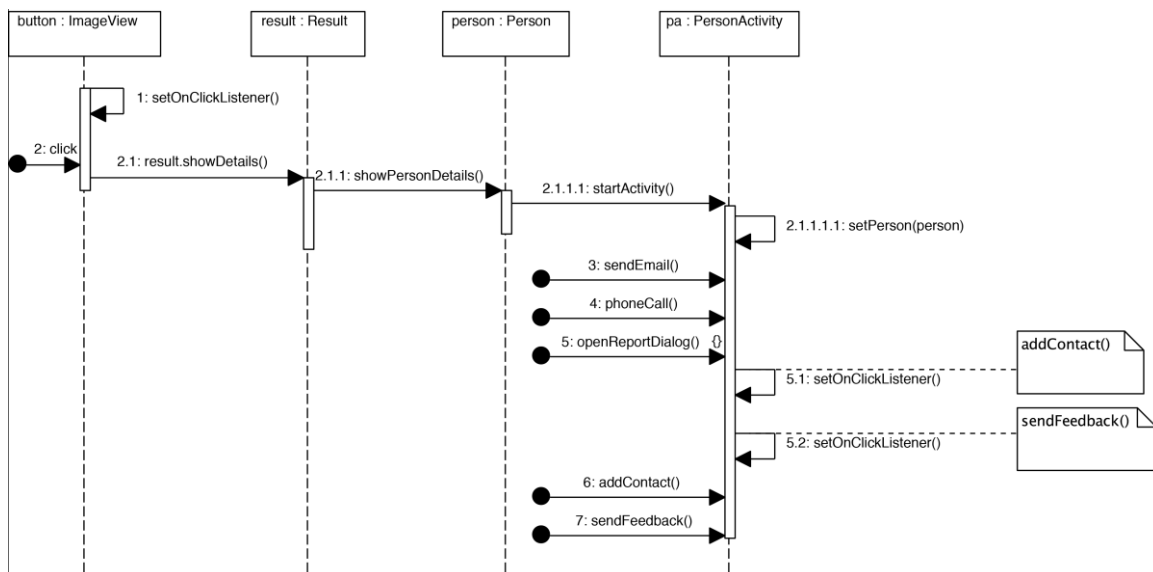


Figura 4.28: Vista de detalle de persona

4.5 DISEÑO DE LA BASE DE DATOS

Como se ha explicado anteriormente, la parte del sistema que se encuentra en el servidor, necesita una base de datos georreferenciada a la cual se accederá a través del servicio Web utilizado por la aplicación *UC3M Mapas*.

La base de datos existía antes del comienzo de este proyecto, pero se cree conveniente explicar su estructura para que el lector sea capaz de comprender la totalidad del sistema.

La estructura de la base de datos se presenta mediante el diagrama Entidad/Relación de la Figura 2.1 que permite al lector ver las entidades con que se trabaja en dicha base de datos, sus atributos y por último las relaciones que existen entre las entidades.

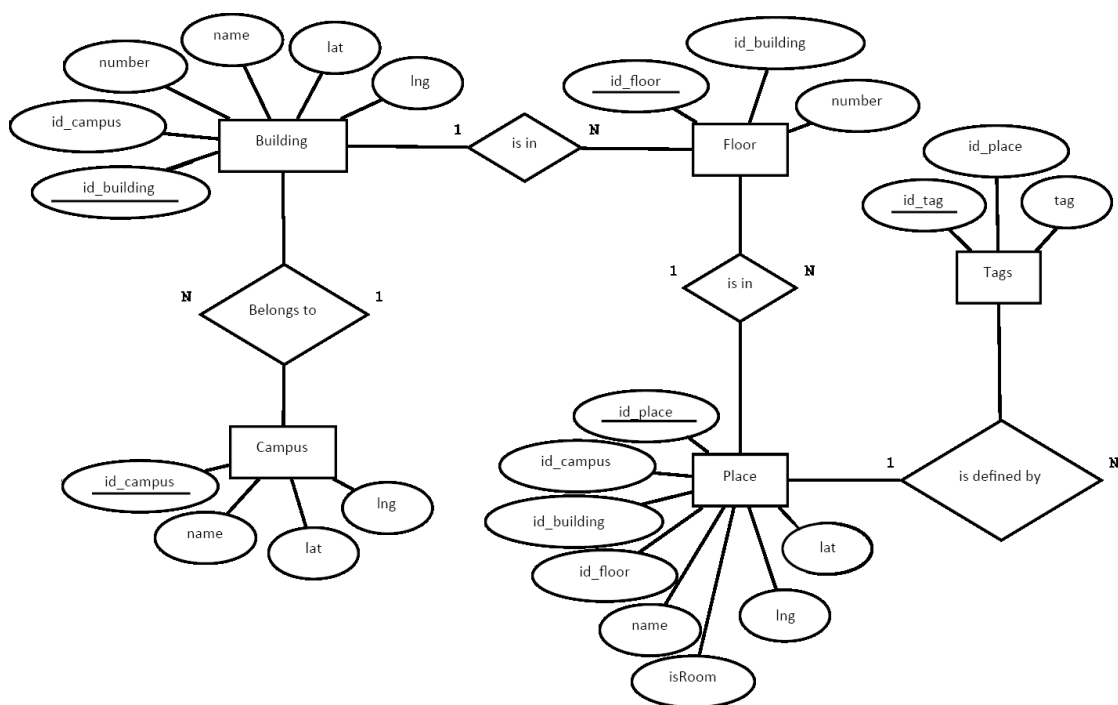


Figura 4.29: Diagrama Entidad-Relación

La transformación a modelo relacional que se ha realizado desde el diagrama Entidad/Relación al modelo relacional para la posterior creación de la base de datos se muestra en la Figura 4.30.

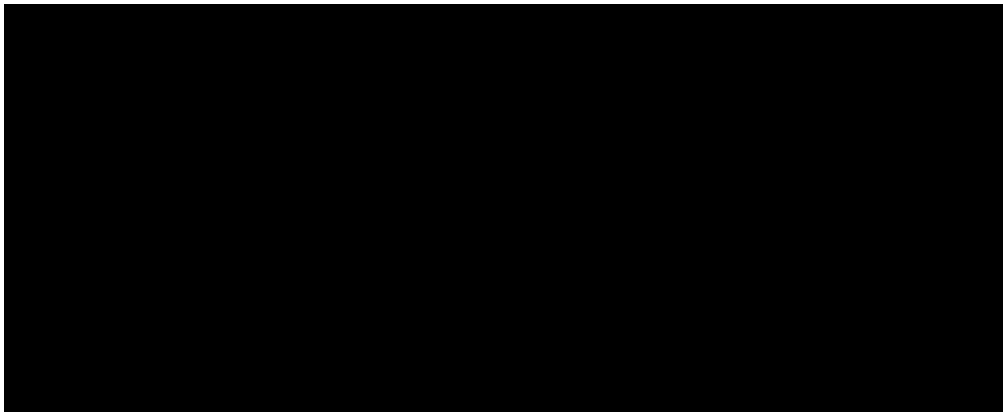


Figura 4.30: Diagrama Relacional

4.6 DISEÑO DEL REPOSITORIO DE PLANOS

Partiendo de los planos originales proporcionados por la universidad en formato *dxf*, se deben modificar con alguna herramienta de edición gráfica de manera que las escaleras de acceso público aparecieran en color verde (con la flecha que indica el sentido en color rojo). Las puertas y rampas también aparecen en el mismo color, de manera que el usuario pueda ver con facilidad cuál es el camino a seguir para llegar a un lugar. El resto de líneas se mostrarán en color negro sobre fondo blanco.

Posteriormente se deben exportar a formato *TIFF* y se deben insertar en un mismo archivo formato *psd* superpuestos sobre una imagen tomada de Google Maps de manera que los planos queden bien orientados.

Los últimos pasos deben ser, como se explica en el artículo del blog de Grimes [16], por un lado exportar cada planta a un fichero *TIFF*, por otro lado, crear un script que obtenga un archivo *VRT* por cada uno de estos *TIFF* y finalmente, utilizar el programa *MapTiler* para generar automáticamente a partir de los *VRT* la estructura de directorios para cada una de las plantas del edificio.

Por último, en el repositorio, los directorios se deben ordenar siguiendo la siguiente estructura:

```
http://<URL del servidor>/tiles/<NombreEdificio>/<n°Planta>/
```

Dentro de estos directorios, la estructura es la indicada en el estándar anteriormente comentado, que consiste en la división por nivel de zoom (utilizando en nuestro caso 17 como zoom mínimo y 21 como zoom máximo) y posteriormente por región. De modo que cuanto mayor sea el nivel de zoom, mayor es el número de “azulejos” que se utilizan para mostrar la misma área.

Este proceso se debe repetir 26 veces, una por cada edificio, y por ello será una de las fases que más esfuerzo requiera a la hora de desarrollar el proyecto.

CAPÍTULO 5 PRUEBAS DE LA APLICACIÓN

5.1 PRUEBAS UNITARIAS	78
5.2 PRUEBAS DE INTEGRACIÓN	85
5.3 PRUEBAS DE SISTEMA	85

En este capítulo se definen los casos de pruebas unitarias, integración y sistema que permiten verificar el correcto funcionamiento de la aplicación.

Las Pruebas de Software, son una serie de procesos de evaluación del software cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto al cliente. Las pruebas de software son una actividad más en el proceso de "aseguramiento de la calidad".

Sin embargo dependiendo del proyecto que se tenga en mente realizar y las metodologías utilizadas en el mismo, las pruebas que se elaboran varían notablemente.

5.1 PRUEBAS UNITARIAS

Para probar el correcto funcionamiento de la aplicación se han diseñado una serie de pruebas unitarias que prueban la funcionalidad de los métodos públicos que existen en cada una de las clases que la conforman.

Siguiendo las recomendaciones de Android, se ha creado un proyecto aparte que sirve como proyecto de prueba. Este proyecto utiliza la *android.test*, una biblioteca que funciona de manera muy similar a la biblioteca *JUnit*.

Lo más interesante es que existen interfaces que permiten probar actividades Android, no sólo clases, de forma que se ejecuta la aplicación en el terminal o en el simulador y de este modo se comprueba en el entorno correcto si las pruebas generan fallos o no.

En primer lugar se ha diseñado una batería de pruebas para la clase *NetworkTester* denominada *NetworkTesterTest*.

A continuación se explica cada una de las pruebas:

- El resultado de la prueba **testHaveNetworkConnection** se considera correcto si el dispositivo está conectado a Internet y fallo en caso contrario.
- El resultado de la prueba **testPreconditions** se considera correcto si tanto el atributo de tipo *NetworkTester* como el atributo de tipo *UC3MMapasActivity* no son nulos.
- El resultado de la prueba **testTestOpenedWiFi** se considera correcto si la red en la que se encuentra el dispositivo no requiere inicio de sesión y fallo en caso contrario.
- El resultado de la prueba **testTestWebservice** se considera correcto si la comunicación con el servicio Web es posible y fallo en caso contrario.

Los resultados obtenidos que se muestran en la Figura 5.1 demuestran que las pruebas se han ejecutado correctamente. Sería conveniente realizar estas pruebas en distintos casos para comprobar todos los posibles resultados dependiendo de la conectividad que se tenga en cada momento.

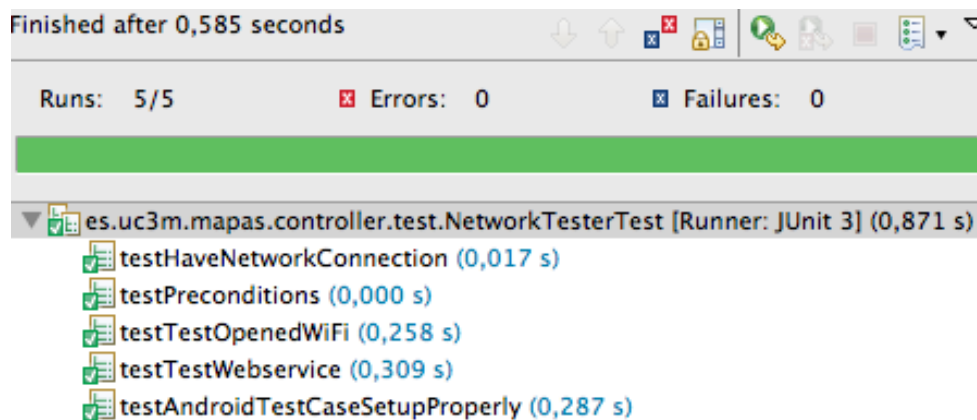


Figura 5.1: Resultados de NetworkTesterTest

La siguiente batería de pruebas se ha realizado sobre la clase *StringHandler* y se denomina *StringHandlerTest*.

- La prueba **testRemoveTilde** se considera correcta si el texto introducido es normalizado de la forma esperada. Para comprobarlo se ha introducido el texto "La brújula señala el Norte" y se ha comparado con el resultado esperado: "La brujula senala el Norte".
- La prueba **testNormalize** se considera correcta si el texto introducido es modificado de la forma esperada. Para comprobarlo se ha introducido el texto "Daniel Peña Sánchez de Rivera" y se ha comparado con el esperado: "Daniel+Pena+Sanchez+de+Rivera".

Los resultados mostrados en la Figura 5.2 indican que las pruebas se han realizado con resultado favorable en ambos casos.

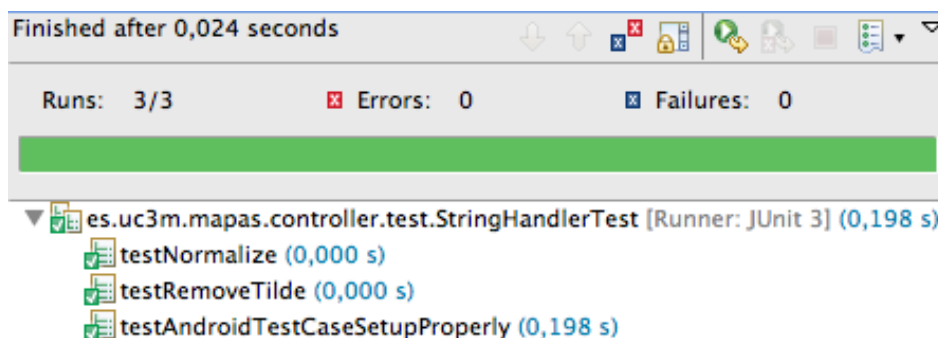


Figura 5.2: Resultados de StringHandlerTest

La siguiente batería de pruebas denominada *SearchTaskTest* es la que se encarga de comprobar el funcionamiento de la clase *SearchTask*.

- El resultado de la prueba **testPreconditions** se considera correcto si el atributo de tipo *SearchTask* no es nulo y fallo en caso contrario.
- El resultado de la prueba **testSearch** se considera correcto si el número de resultados devueltos para una consulta concreta es el esperado y fallo en caso contrario. En este caso, se ha probado a buscar personas en el campus de Leganés utilizando el término "sanchez+hernandez". Teniendo en

cuenta que en el momento en que se realizó la prueba sólo había dos trabajadores con esos apellidos en el campus, la comparación del número de resultados se hizo con el número 2.

Como se puede observar en la Figura 5.3, los resultados de ambas pruebas son correctos.

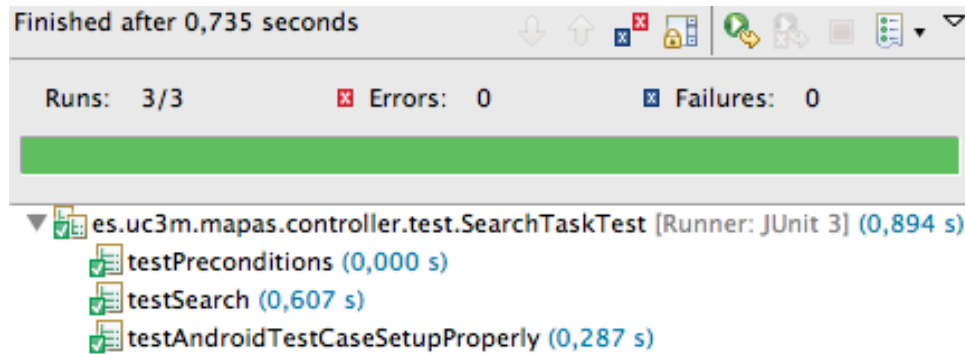


Figura 5.3: Resultados SearchTaskTest

Siguiendo con las pruebas unitarias, se ha diseñado la batería de pruebas denominada *PersonTest* con las que se prueban las funcionalidades de la clase *Person*.

Para la realización de esta prueba, en el constructor se ha creado un objeto de tipo *Person* con una serie de parámetros con los que se harán las comparaciones en las diferentes pruebas.

A continuación se explica brevemente cada una de las pruebas que conforman esta batería:

- El resultado de las pruebas del estilo **testGet...** se considera correcto si el edificio obtenido es igual al esperado.
- El resultado de la prueba **testPerson** se considera correcto si el método se puede ejecutar sin fallos y se considera fallo en caso contrario.

Como se puede comprobar en la Figura 5.4, los resultados de todas las pruebas que conforman esta batería son correctos.

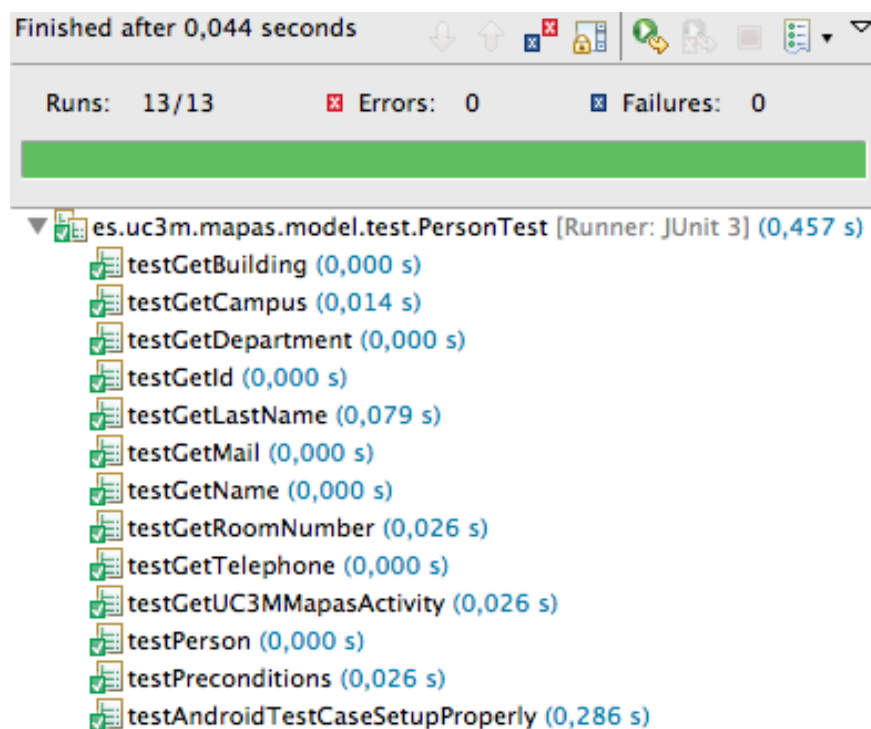


Figura 5.4: Resultados de PersonTest

Las pruebas definidas en la batería de pruebas denominada *PlaceTest*, que prueba la funcionalidad de la clase *Place*, son muy similares a las realizadas en la batería de pruebas *PersonTest*, por lo que se considera que no procede explicarlas.

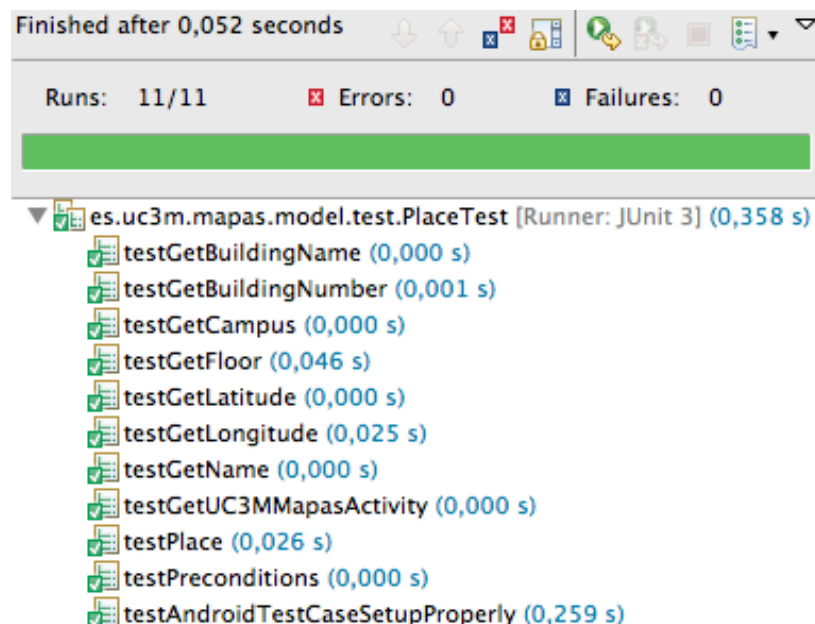


Figura 5.5: Resultados de PlaceTest

Las pruebas que conforman la batería de pruebas denominada *UC3MMapasActivityTest* prueban la funcionalidad de la clase *UC3MMapasActivity*, que es la principal del proyecto.

- El resultado de la prueba **testApi** se considera correcto en caso de que ...
- El resultado de la prueba **testGetResultsFromServer** se considera correcto en caso de que el número de resultados devueltos sea el esperado. Para probarlo, se ha buscado una persona en Leganés con el siguiente término de búsqueda: "**Javier%20Sanchez%20Hernandez**". Como es de esperar, el resultado devuelto es sólo uno. Para probar la búsqueda de lugares, se ha buscado también en Leganés con el término "**2.2.B10**" y el resultado esperado es 2, ya que al buscar con ese término se obtienen como resultados el aula 2.2.B10 y el aula 2.2.B10Izq.
- El resultado de la prueba **testIsApplicationSentToBackground** se considera correcto en caso de que la aplicación no esté en segundo plano y fallo en caso contrario.
- El resultado de la prueba **testLoadSettings** se considera correcto en caso de que las preferencias se carguen sin problema, es decir, que se acceda bien al archivo de preferencias para obtener los valores en él indicados. En caso contrario, se considera fallo.
- El resultado de la prueba **testPreconditions** se considera correcto si el atributo de la clase *UC3MMapasActivity* utilizado para la realización de las demás pruebas no es nulo. Se considera fallo en caso contrario.
- El resultado de la prueba **testSetCampus** se considera correcto en el caso de que el campus seleccionado sea el esperado después de la ejecución. Se considera fallo en caso contrario.
- El resultado de la prueba **testSetSearchType** se considera correcto en el caso de que el tipo de búsqueda seleccionado sea el esperado después de la ejecución. Se considera fallo en caso contrario.
- El resultado de la prueba **testSettingsListener** se considera correcto si durante la ejecución de la prueba se muestra en el dispositivo la vista de configuración de la aplicación. Se considera fallo en caso contrario.
- El resultado de la prueba **testShowAbout** se considera correcto si durante la ejecución de la misma se muestra en el dispositivo la vista que muestra la información acerca de la aplicación. Se considera fallo en caso contrario.
- El resultado de la prueba **testShowHelp** se considera correcto si durante la ejecución de la misma se muestra en la pantalla del dispositivo la vista de ayuda que explica al usuario cómo utilizar la aplicación. Se considera fallo en caso contrario.
- El resultado de la prueba **testShowMap** se considera correcta si durante la ejecución de la misma se muestra en la pantalla del dispositivo el mapa con los parámetros correctos (obtenidos de la configuración). Se considera fallo en caso contrario.

En la Figura 5.6 se muestran los resultados obtenidos de las pruebas anteriormente explicadas.

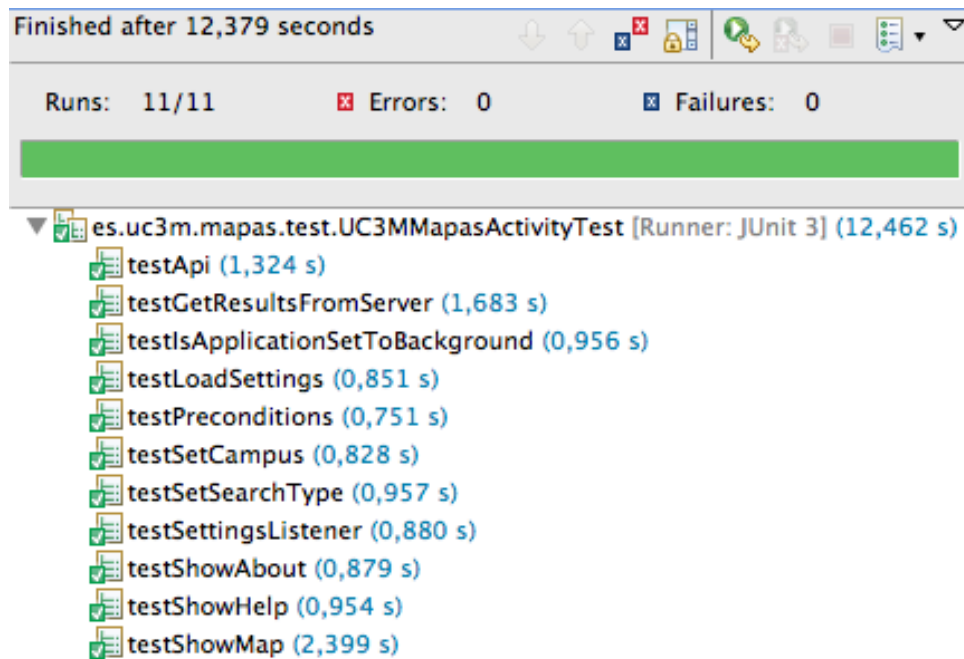


Figura 5.6: Resultados de UC3MMapasActivityTest

Las pruebas que conforman la batería de pruebas denominada *AboutActivityTest* prueban la funcionalidad de la clase *AboutActivity*, que es la que se encarga de mostrar la información de la aplicación.

A continuación se describen las diferentes pruebas realizadas:

- La prueba **testIsApplicationSentToBackground** es igual que la realizada en la prueba de mismo nombre que se encuentra dentro de la batería *UC3MMapasActivityTest*.
- El resultado de la prueba **testPreconditions** se considera correcto si el atributo del tipo *AboutActivity* no es nulo y fallo en caso contrario.
- El resultado de la prueba **testSendEmail** se considera correcto si durante la ejecución de la misma, la aplicación de correo electrónico se abre de manera que el usuario pueda enviar un correo electrónico a la dirección especificada en la actividad. El resultado se considera fallo en caso contrario.
- El resultado de la prueba **testShare** se considera correcto si durante la ejecución de la misma, se abre el diálogo que permite al usuario seleccionar el servicio a través del cual se puede compartir el texto. En caso contrario, el resultado se considera fallo.
- El resultado de la prueba **testShowCredits** se considera correcto si durante la ejecución de la misma, se muestra en la pantalla del dispositivo la vista que muestra los créditos de la aplicación. En caso contrario, el resultado se considera fallo.

En la Figura 5.7 se muestran los resultados de las diferentes pruebas realizadas en esta batería.

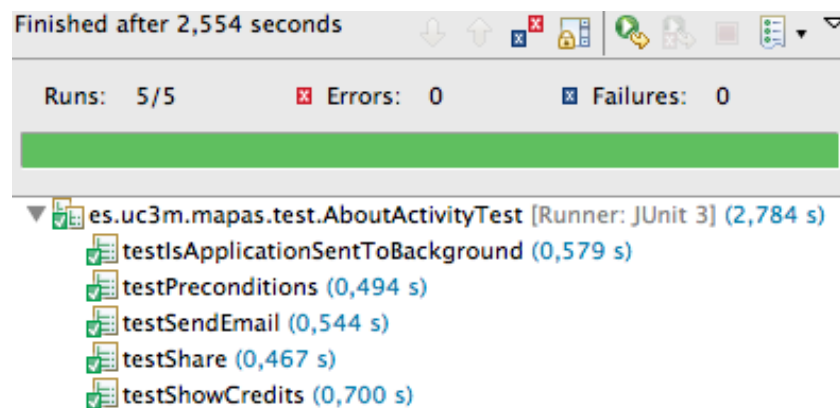


Figura 5.7: Resultados de AboutActivityTest

Por último, se ha desarrollado una batería de pruebas denominada *ListItemTest* que prueba la funcionalidad de la clase *ListItem* mediante una serie de pruebas explicadas a continuación.

- El resultado de la prueba **testGetCategory** se considera correcto si la categoría obtenida es igual a la esperada. Se considera fallo en caso contrario.
- El resultado de la prueba **testGetSelectedValue** se considera correcto si el valor seleccionado es igual al esperado y fallo en caso contrario.
- El resultado de la prueba **testSetCategory** se considera correcto si tras la ejecución de la prueba, el valor de la categoría ha cambiado. Se considera fallo en caso contrario.
- El resultado de la prueba **testSetSelectedValue** se considera correcto si tras la ejecución de la prueba, el valor seleccionado ha cambiado. Se considera fallo en caso contrario.

En la Figura 5.8 se muestran los resultados tras ejecutar la batería de pruebas. Como se puede observar, todas ellas han sido correctas.

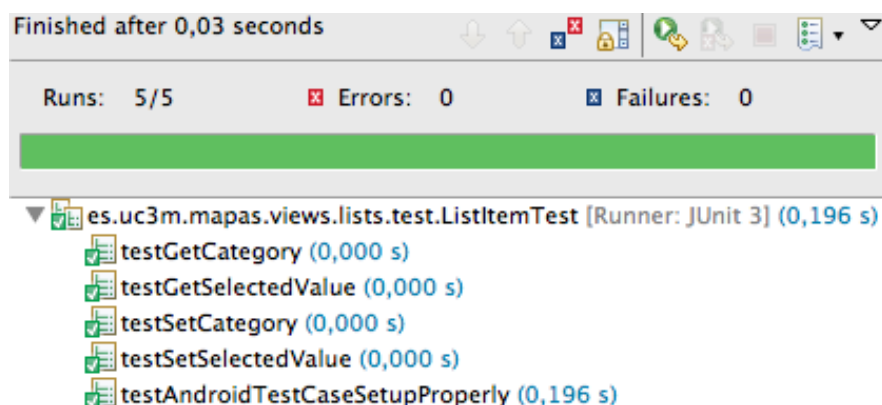


Figura 5.8: Resultados de ListItemTest

5.2 PRUEBAS DE INTEGRACIÓN

La aplicación *UC3M Mapas* en su versión Android se comunica con un servidor mediante un servicio Web que devuelve datos en formato *JSON*.

Al haberse hecho la integración desde el inicio del desarrollo no se considera necesario realizar pruebas de integración en este sentido.

Tampoco se realizarán pruebas de integración con la biblioteca *osmdroid* puesto que al igual que en el caso anterior, se ha integrado desde prácticamente el inicio del desarrollo y su integración está suficientemente probada.

5.3 PRUEBAS DE SISTEMA

Dado que se han utilizado técnicas ágiles de desarrollo para el proyecto. En las pruebas también se seguirá la misma metodología. Siendo el cliente el que valore si el proyecto avanza como desea. Para ello se debe cuidar el cumplimiento de los siguientes puntos para que el resultado de las pruebas se el óptimo:

- El cliente debe tomar parte activa en la creación de las pruebas de sistema.
- En muchas ocasiones no se realizan todas las pruebas de sistema hasta que no se entrega el software al cliente. Las pruebas de aceptación se deben proporcionar en cada iteración (lo deseable es que estén listas a la mitad de la misma).
- Únicamente se deben probar las cosas que quiere que funcionen.
- Es imprescindible especificar pruebas de sistema para cada una de las historias que forman las iteraciones.

Cada una de las pruebas deberá ser especificada por una serie de campos que se muestran a continuación. A través de una plantilla que servirá como modelo para el resto de las pruebas.

- **ID:** el identificador de la prueba estará compuesto por dos valores separados por un punto. El primero de ellos identifica el diagrama e secuencia al que se refiere. El segundo identifica el número de prueba para ese diagrama de secuencia.
- **Descripción:** Es el objetivo que tiene marcado la prueba para que sea pasada.
- **Pasos:** Son la serie de pasos que se ha de hacer en la aplicación para poder realizar la prueba.
- **Errores posibles:** Se enumeran los posibles errores que se han encontrado o pueden encontrarse al realizar la prueba, y que no permiten que esté pasada.
- **Requisitos:** Se enumeran los requisitos con los que está relacionada la prueba.
- **Estado:** Tras realizar la prueba es el resultado de la misma “pasada” si es positivo y “no pasada” si es negativo.

ID
Descripción
Pasos
Errores posibles
Requisitos
Estado

Tabla 5.1: Plantilla de prueba

A continuación en cada uno de los apartados siguientes se pasa a especificar las pruebas que se han realizado sobre cada uno de los **diagramas de secuencia** establecidos para el proyecto en el capítulo anterior. Además se han añadido algunas pruebas para probar otra funcionalidad no descrita en estos diagramas.

La primera de ellas, especificada en la Tabla 5.2 prueba que el inicio de la aplicación sea correcto, es decir: la aplicación deberá iniciarse sin que se muestre ninguna advertencia acerca de la conexión de red o de la conexión con el servicio web.

ID	P1.1
Descripción	La aplicación deberá iniciarse
Pasos	1. Arrancar la aplicación.
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet. • El dispositivo está conectado a una red wifi que requiere inicio de sesión. • La conexión con el servidor no es posible.
Requisitos	RR-001, RI-008
Estado	PASADA

Tabla 5.2: P1.1 - La aplicación deberá iniciarse

En la Tabla 5.3 se define la prueba que se encarga de comprobar si la aplicación ha sido abierta por otra aplicación Android. En ese caso, la aplicación debe recibir los parámetros enviados por la otra aplicación y mostrar el resultado de la consulta recibida en la pantalla principal.

ID	P1.2
Descripción	Recibir parámetros desde otra aplicación
Pasos	<ol style="list-style-type: none">1. Arrancar una aplicación de prueba.2. Definir los valores de los parámetros a enviar.3. Llamar a la aplicación con esos parámetros.
Errores posibles	<ul style="list-style-type: none">• Que falte algún parámetro.• Que alguno de los valores de los parámetros a enviar no sea correcto.• Que el parámetro de búsqueda no se encuentre en el campus indicado.
Requisitos	RI-001
Estado	PASADA

Tabla 5.3: P1.2 - Recibir parámetros desde otra aplicación

La tercera prueba se define en la Tabla 5.4 y consiste al igual que la anterior en comprobar si la aplicación es capaz de recibir parámetros a través de una URL.

Para comprobarlo, se intenta abrir la URL en el navegador del terminal, y si nuestra aplicación aparece como posible para abrir el enlace, significa que funciona. Al abrirla, se mostrará el resultado teniendo en cuenta los parámetros de la URL.

ID	P1.3
Descripción	Recibir parámetros desde una URL
Pasos	<ol style="list-style-type: none">1. Escanear un código QR o teclear en el navegador una URL.2. Elegir la aplicación entre las posibles para abrir el enlace.
Errores posibles	<ul style="list-style-type: none">• Que falte algún parámetro.• Que alguno de los valores de los parámetros a enviar no sea correcto.• Que el parámetro de búsqueda no se encuentre en el campus indicado.
Requisitos	RI-002
Estado	PASADA

Tabla 5.4: P1.3 - Recibir parámetros desde una URL

La siguiente prueba consiste en comprobar que la aplicación es capaz de encontrar los lugares que busca el usuario. En la Tabla 5.5 se muestra el procedimiento realizado y el resultado.

La búsqueda se ha realizado en Leganés con el término “2.1.C02” tal y como se muestra en el texto de ejemplo.

ID	P2.1
Descripción	Búsqueda de un lugar
Pasos	<ol style="list-style-type: none"> 1. Seleccionar el campus en el que se desea realizar la búsqueda (en caso de ser diferente al ya establecido). 2. Seleccionar el tipo de búsqueda de lugar. 3. Introducir el término de búsqueda. 4. Pulsar el botón de búsqueda.
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet. • El dispositivo está conectado a una red wifi que requiere inicio de sesión. • La conexión con el servidor no es posible. • La localización que busca no está georreferenciada en la base de datos.
Requisitos	RI-003, RI-004, RI-008, RF-001, RF-002, RF-012, RR-001
Estado	PASADA

Tabla 5.5: P2.1 - Búsqueda de un lugar

Después de buscar este lugar, se ha probado la funcionalidad que permite reportar incidencias al administrador. En la Tabla 5.6 se muestra el proceso llevado a cabo para reportar esta incidencia y el resultado obtenido.

ID	P2.2
Descripción	Reportar una incidencia de un lugar
Pasos	<ol style="list-style-type: none">1. Seleccionar el campus en el que se desea realizar la búsqueda (en caso de ser diferente al ya establecido).2. Seleccionar el tipo de búsqueda de lugar.3. Introducir el término de búsqueda.4. Pulsar el botón de búsqueda.5. Pulsar el botón de la burbuja de información para ver detalles del lugar.6. Pulsar el botón de la parte superior de la pantalla para ver las opciones posibles.7. Elegir la opción de reportar incidencia.8. Rellenar el correo electrónico con los datos pedidos y enviarlo.9. Cerrar la aplicación.
Errores posibles	<ul style="list-style-type: none">• El dispositivo no tiene conexión a Internet.• El dispositivo está conectado a una red wifi que requiere inicio de sesión.• La conexión con el servidor no es posible.• La localización que busca no está georreferenciada en la base de datos.• El lugar no tiene detalles a mostrar.
Requisitos	RF-003, RF-005
Estado	PASADA

Tabla 5.6: P2.2 Incidencia de lugar

La siguiente prueba consiste en la búsqueda de una persona en el campus de Getafe. El término de búsqueda introducido para la prueba ha sido “daniel peña”.

El resultado será un marcador en el punto en el que se encuentra el despacho del rector de la Universidad Carlos III de Madrid.

En la Tabla 5.7 se muestran tanto el procedimiento seguido como el resultado obtenido tras la realización de la prueba.

ID	P2.3
Descripción	Búsqueda de una persona
Pasos	<ol style="list-style-type: none"> 1. Seleccionar el campus en el que se desea realizar la búsqueda (en caso de ser diferente al ya establecido). 2. Seleccionar el tipo de búsqueda de persona. 3. Introducir el término de búsqueda. 4. Pulsar el botón de búsqueda.
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet. • El dispositivo está conectado a una red wifi que requiere inicio de sesión. • La conexión con el servidor no es posible. • La persona que busca no se encontró en el campus seleccionado. • El despacho de la persona obtenida no está georreferenciado en la base de datos.
Requisitos	RI-003, RI-004, RI-008, RF-001, RF-002, RF-012, RR-001
Estado	PASADA

Tabla 5.7: P2.3 Búsqueda de una persona

La siguiente prueba definida en la Tabla 5.8 consiste en probar la funcionalidad de todas las acciones disponibles desde la vista de detalle de persona.

Estas posibles acciones se enumeran a continuación:

- Reportar incidencia
- Agregar a contactos
- Enviar correo electrónico
- Llamar por teléfono

ID	P2.4
Descripción	Ejecutar todas las posibles acciones desde la vista de detalle de persona
Pasos	<ol style="list-style-type: none">1. Seleccionar el campus en el que se desea realizar la búsqueda (en caso de ser diferente al ya establecido).2. Seleccionar el tipo de búsqueda de persona.3. Introducir el término de búsqueda.4. Pulsar el botón de búsqueda.5. Pulsar el botón de la burbuja de información para ver detalles de la persona.6. Pulsar el botón de la parte superior de la pantalla para ver las opciones posibles.7. Elegir la opción de reportar incidencia.8. Rellenar el correo electrónico con los datos pedidos y enviarlo.9. Volver a la vista de detalle.10. Pulsar sobre el correo electrónico mostrado.11. Redactar el correo y enviarlo.12. Volver a al vista de detalle.13. Pulsar sobre el número de teléfono mostrado.14. Llamar a la persona.15. Volver a la vista de detalle.16. Pulsar el botón de la parte superior de la pantalla para ver las opciones posibles.17. Seleccionar la opción de agregar a contactos.18. Guardar el contacto.19. Volver a la vista de detalle.20. Cerrar la aplicación.
Errores posibles	<ul style="list-style-type: none">• El dispositivo no tiene conexión a Internet.• El dispositivo está conectado a una red wifi que requiere inicio de sesión.• La conexión con el servidor no es posible.• La persona que busca no se encontró en el campus seleccionado.• El despacho de la persona obtenida no está georreferenciado en

	<p>la base de datos.</p> <ul style="list-style-type: none"> • La aplicación de correo electrónico falla. • La aplicación de teléfono falla o no está disponible. • La aplicación de contactos falla.
Requisitos	RF-004, RF-005, RF-006, RF-007, RF-008
Estado	PASADA

Tabla 5.8: P2.4 Funcionalidad de la vista de detalle de persona

En la siguiente prueba definida en la Tabla 5.9 se comprueba la correcta visualización de los diferentes tipos de mapa posibles desde la configuración de la aplicación.

Los únicos tipos de vista de mapa en este caso son: normal y satélite.

ID	P2.5
Descripción	Comprobar que se muestran correctamente los diferentes tipos de mapa
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Acceder a la pantalla de configuración. 3. Elegir tipo de mapa normal. 4. Volver a la pantalla principal.
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet. • El dispositivo está conectado a una red wifi que requiere inicio de sesión. • La conexión con el servidor no es posible.
Requisitos	RI-003, RI-004, RF-013
Estado	PASADA

Tabla 5.9: P2.5 Tipo de mapa

En la Tabla 5.10 se define la prueba que se encarga de comprobar la aplicación genera un aviso al usuario en caso de que el servidor no responda en un tiempo determinado.

Para ello se desconecta el servidor, de manera que la conexión con él sea imposible.

ID	P3.1
Descripción	En caso de que el tiempo de respuesta del servidor sobrepase el tiempo máximo de consulta, se mostrará un diálogo explicando que no se ha podido realizar la conexión con el servidor.
Pasos	<ol style="list-style-type: none">1. Iniciar la aplicación.2. Desconectar el servidor.3. Realizar una búsqueda de persona o lugar.
Errores posibles	<ul style="list-style-type: none">• El dispositivo no tiene conexión a Internet.• El dispositivo está conectado a una red wifi que requiere inicio de sesión.
Requisitos	RR-001
Estado	PASADA

Tabla 5.10: P3.1 Tiempo máximo de respuesta

En la siguiente prueba definida en la Tabla 5.11 se comprueba si la aplicación detecta que el dispositivo está conectado a una red pública que requiere inicio de sesión y en este caso, si redirecciona de forma correcta al usuario para que inicie sesión desde su navegador.

ID	P3.2
Descripción	En caso de que el dispositivo esté conectado a una red inalámbrica pública que requiera inicio de sesión mediante aceptación de condiciones, durante el inicio de la aplicación se abrirá la aplicación de navegador mostrando la página Web en la que el usuario debe aceptar los términos.
Pasos	<ol style="list-style-type: none">1. Conectar el dispositivo a una red pública que requiere inicio de sesión mediante aceptación de condiciones.2. Iniciar la aplicación.
Errores posibles	<ul style="list-style-type: none">• El dispositivo no tiene conexión a Internet.
Requisitos	RR-002
Estado	PASADA

Tabla 5.11: P3.2 Red inalámbrica pública

En la Tabla 5.12 se comprueba la funcionalidad de todas las acciones que se pueden realizar desde la vista de acerca de que ofrece la aplicación.

ID	P4.1
Descripción	El usuario entrará en la vista de acerca de y probará cada una de las opciones que en ella se encuentran
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Pulsar el botón de menú. 3. Elegir la opción Acerca de en el menú de opciones. 4. Pulsar sobre el correo electrónico para que se abra la aplicación de correo electrónico. 5. Pulsar el botón de volver atrás. 6. Pulsar sobre el texto que dice “Compartir utilizando” 7. Elegir alguna de las aplicaciones ofrecidas para compartir la información. 8. Compartir texto. 9. Pulsar el botón de volver atrás. 10. Pulsar sobre el nombre de la cuenta de twitter. 11. Volver atrás mediante el botón. 12. Mostrar los créditos. 13. Volver atrás. 14. Volver atrás. 15. Cerrar la aplicación.
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet. • El dispositivo está conectado a una red wifi que requiere inicio de sesión. • La conexión con el servidor no es posible. • La aplicación de correo electrónico falla. • La aplicación elegida para compartir texto falla. • Error al conectar a twitter.
Requisitos	RI-005, RI-006, RF-009, RF-010, RF-011
Estado	PASADA

Tabla 5.12: P4.1 Acerca de

La última prueba realizada y definida en la Tabla 5.13 comprueba que la ayuda al usuario se muestra de forma correcta, de manera que la navegación entre las diferentes pantallas sea correcta.

ID	P5.1
Descripción	La vista de ayuda debe mostrar en cada una de sus pantallas dos flechas de navegación, excepto en la primera y la última, en las que únicamente aparecerá una.
Pasos	<ol style="list-style-type: none">1. Iniciar la aplicación.2. Pulsar el botón de menú.3. Elegir la opción de ayuda.4. Navegar por la vista haciendo uso de los botones mostrados en la parte inferior.5. Pulsar el botón de volver, para ir la vista de inicio otra vez.6. Cerrar la aplicación.
Errores posibles	<ul style="list-style-type: none">• El dispositivo no tiene conexión a Internet.• El dispositivo está conectado a una red wifi que requiere inicio de sesión.• La conexión con el servidor no es posible.
Requisitos	RI-007
Estado	PASADA

Tabla 5.13: P5.1 Ayuda al usuario

CAPÍTULO 6 CONCLUSIONES Y LÍNEAS FUTURAS

6.1	CONCLUSIONES.....	98
6.2	LÍNEAS FUTURAS.....	100
6.2.1	MEJORA EN LA LOCALIZACIÓN EN INTERIORES.....	100
6.2.2	NUEVOS EDIFICIOS Y CAMPUS	101
6.2.3	GEOLOCALIZACIÓN DE LIBROS EN LAS BIBLIOTECAS DE LA UNIVERSIDAD	101
6.2.4	RED SOCIAL PARA LOCALIZACIÓN DE AMIGOS.....	101
6.2.5	GEOLOCALIZACIÓN DE SERVICIOS UNIVERSITARIOS.....	101
6.2.6	GEOLOCALIZACIÓN DE DESFIBRILADORES	101
6.2.7	CÓMO LLEGAR	102
6.2.8	INFORMACIÓN DE AULAS.....	102

En este capítulo se incluyen las conclusiones obtenidas durante el desarrollo del proyecto, entre las que se encuentra una evaluación personal de las experiencias y aprendizaje adquiridos durante la realización del PFC, así como las líneas futuras para la ampliación y mejora de la aplicación.

6.1 CONCLUSIONES

Se ha desarrollado como proyecto de fin de carrera una aplicación móvil de geolocalización en los campus de la Universidad Carlos III de Madrid que permite al usuario encontrar tanto lugares como personas dentro de los recintos de la universidad así como la consulta de los detalles de las personas o lugares buscados y la interacción con ellos. Se ha elegido la plataforma Android puesto que es la que mayor número de usuarios tiene en España y además el porcentaje de usuarios de esta plataforma en comparación con las demás está creciendo en los últimos meses. Es una aplicación muy útil que permite que el usuario pueda orientarse dentro de los distintos campus de la universidad de manera que pueda encontrar con facilidad y de forma rápida lo que busca, tanto lugares como personas. Además, la búsqueda de personas ofrece datos de contacto de esas personas que podrán ser utilizados en caso de que no sea posible llegar hasta el despacho en concreto por alguna razón. Además permite seleccionar el modo de vista del mapa (normal o satélite), el campus a mostrar y el tipo de búsqueda (lugar o persona).

La aplicación, denominada *UC3M Mapas*, está disponible desde el día 23 de abril de 2012 en Google Play (la tienda online de aplicaciones Android) a través del siguiente enlace:

<https://play.google.com/store/apps/details?id=es.uc3m.mapas>

Durante el desarrollo de este proyecto he puesto en práctica muchos de los conocimientos adquiridos en las diferentes asignaturas cursadas durante los cinco cursos de la carrera, sobre todo de las asignaturas orientadas al desarrollo de software y programación orientada a objetos en las que he trabajado principalmente con el lenguaje de programación Java. Además de para aplicar estos conocimientos, el desarrollo de este proyecto me ha servido para aprender nuevos conceptos acerca de la geolocalización utilizando la API de Google Maps, con la cual ya había tratado en un trabajo dirigido realizado en mayo de 2009 que servía para localizar al médico especialista más cercano para tratar la enfermedad que tenía el usuario. Para ello se hacía uso del cálculo de rutas a pie o en automóvil para obtener los tiempos y hacer comparaciones. También me ha servido para aprender cuál debe ser el orden que se debe seguir a la hora de desarrollar aplicaciones móviles Android ya que aunque los tutoriales que se pueden encontrar por Internet explican de manera sencilla cómo realizar aplicaciones pequeñas con funcionalidades concretas, cuando uno se enfrenta a problemas más grandes, no se pueden aplicar los mismos criterios.

La realización de este PFC me ha parecido muy interesante debido a que en este momento las aplicaciones móviles están en auge y en concreto, en España, Android está creciendo en estos últimos meses de manera considerable frente a otras plataformas. La parte positiva de esta situación es que sabiendo desarrollar este tipo de aplicaciones y con buenas ideas uno puede entrar a un mercado por un precio de tan solo 25 \$. Durante este periodo he comprobado que lanzar una aplicación al mercado implica mucho más que el desarrollo. Para obtener buenos resultados de ventas/descargas será necesario

elaborar una estrategia de marketing que permita actuar en consecuencia para obtener más ganancias.

En este proceso no todo han sido éxitos. Ha habido momentos en los que han surgido problemas. La mayor parte de ellos se han solucionado de una forma u otra. Cabe destacar el problema de la localización. Al inicio se hacía de una forma bastante complicada que utilizaba GPS en caso de que estuviera conectado y si no, utilizaba redes inalámbricas. Se solucionó utilizando la clase *MyLocationOverlay* ya que ella misma se encarga de buscar las fuentes de las que obtiene la posición. También fue complicado integrar la biblioteca *osmdroid*, pero con un poco de investigación fue posible y por supuesto muy útil.

También existe un problema con la nomenclatura de las aulas y despachos. Durante la fase de georreferenciación, se ha utilizado la nomenclatura estándar de la universidad, pero al realizar pruebas, se han encontrado problemas ya que un aula georreferenciada aparecía en la aplicación no georreferenciada. Investigando un poco se dedujo que el problema está en que el directorio de la universidad maneja algunos identificadores de manera diferente a la nuestra en caso de salas que utilizan la palabra latina *bis* en su parte final.

Por último, cabe destacar un problema cuya solución no es fácil puesto que depende de la voluntad de todos los miembros de la comunidad universitaria. El problema reside en que cada persona debe hacerse responsable de que sus datos estén actualizados en el directorio de manera que cuando alguien busque a través de la aplicación, los datos sean correctos.

Por otro lado, al inicio del proyecto se planteó que su duración fuera de 4 meses, pero finalmente ha sido de 9 meses. Esto ha sido debido entre otras cosas a que en los primeros meses no se ha podido dedicar al proyecto tiempo completo y a que la fecha de lanzamiento de la primera versión se retrasó durante el desarrollo del proyecto hasta el día 23 de abril de 2012. Después de esta fecha se hicieron algunas mejoras en la aplicación y se desarrolló la pantalla de ayuda. Además, fue a partir de este momento cuando se comenzó esta memoria.

El resultado final tras el desarrollo de la aplicación a nivel personal ha sido ampliamente gratificante, proporcionando una serie de conocimientos que no se habían adquirido durante la carrera y que pueden ser de utilidad para la vida laboral en un futuro cercano.

En lo que respecta a los objetivos que se habían establecido para este PFC, se evaluará hasta qué punto se ha llegado a la consecución de los mismos. Para ello, se volverán a enumerar cada uno de ellos, adjuntando las conclusiones que se pueden obtener tras el desarrollo:

- **Desarrollar servicios que faciliten la orientación a los usuarios dentro de los diferentes campus de la UC3M:** Este objetivo era el objetivo general y se ha visto cumplido gracias al cumplimiento de los objetivos concretos explicados en los siguientes puntos.
- **Desarrollar de un servicio de localización para la UC3M:** Este objetivo se concluyó durante las primeras semanas del proyecto, puesto que es la base para poder realizar todos los demás objetivos. Consistió en el

desarrollo de la base de diseñada por el grupo de investigación SEL-UC3M, el servicio Web que permitía la actualización de esa base de datos y una aplicación Web que hacía uso de ese servicio para poder consultar, insertar y modificar datos georreferenciados en la base de datos.

- **Adaptación de los planos de los edificios:** Antes de poder georreferenciar los campus fue necesario adaptar los planos de los edificios para poder mostrarlos superpuestos en el plano de *Google Maps* que ofrece la aplicación. Se han realizado planos para los 26 edificios que en este momento están en funcionamiento.
- **Georreferenciación de los campus de la UC3M:** Según se iban adaptando los planos de un edificio y se subían al servidor, se georreferenciaban todas las salas que tuvieran un identificador y se actualizaba la base de datos, de manera que una vez se ha finalizado el proceso y gracias a la herramienta Web para poder insertar lugares en la base de datos desarrollada por el grupo de investigación SEL-UC3M, ha sido posible insertar un total de 2125 lugares repartidos por los 26 edificios de los 3 campus de que dispone la Universidad Carlos III de Madrid.
- **Desarrollo de una aplicación Android:** Este objetivo se ha cumplido, puesto que la aplicación cumple con todos los requisitos explicados en la fase de diseño y permite al usuario acceder a los servicios de localización proporcionados por los demás objetivos.

También se ha entendido que la aplicación debe ser lo más ligera y veloz posible por lo que se ha tratado de optimizar el código en la medida de lo posible. Además se ha comentado el código Java mediante comentarios Javadoc que permiten que otros programadores comprender la estructura de la aplicación y sean capaces de modificarla para futuras mejoras.

6.2 LÍNEAS FUTURAS

Una vez expuestas las conclusiones, se pueden obtener una serie de líneas futuras que no se extraen de los objetivos pero que sería interesante desarrollar. A continuación se muestran las líneas futuras para el proyecto *UC3M Mapas*.

6.2.1 MEJORA EN LA LOCALIZACIÓN EN INTERIORES

La localización en interiores (indoor) es una parte fundamental en esta aplicación ya que la información acerca de dónde se encuentra el usuario es fundamental para saber llegar a los lugares indicados sobre el mapa.

En la versión actual, la localización se obtiene de varias fuentes. Por un lado, se utiliza la señal telefónica, de manera que la localización será más correcta si se utiliza una tarjeta SIM en el dispositivo. Por otro lado, se utilizan las señales de los routers wifi geolocalizados por Google y por último, en caso de que esté activado el receptor en el dispositivo, los satélites GPS.

El problema es que en interiores, estos métodos no funcionan todo lo bien que se podría esperar, por lo que en futuras versiones de la aplicación parece necesario incluir

otras técnicas de localización en interiores como puede ser triangulación con redes wifi o tarjetas *RFID* en puntos estratégicos de manera que la localización se vaya actualizando cuando el usuario pase por esos puntos.

Una solución más sencilla puede consistir en la colocación de códigos QR en puntos concretos de los campus que permitan a la aplicación localizar al usuario sobre el plano. A pesar de ser más rudimentaria que las anteriores, sirve perfectamente para solucionar el problema del posicionamiento en interiores.

6.2.2 NUEVOS EDIFICIOS Y CAMPUS

Actualmente hay varios edificios en construcción o que todavía no han sido abiertos al público aunque no tardarán en hacerlo, por lo que habrá que realizar todo el proceso con cada uno de ellos. Estos nuevos edificios se encuentran en los campus de Leganés, Getafe y en el nuevo campus de Puerta de Toledo.

6.2.3 GEOLOCALIZACIÓN DE LIBROS EN LAS BIBLIOTECAS DE LA UNIVERSIDAD

Teniendo en cuenta que existe una aplicación que permite georreferenciar lugares en los campus universitarios, es muy sencillo geolocalizar por ejemplo cada una de las estanterías que se encuentran en las bibliotecas, de modo que cuando un usuario busque un libro, se le indique por medio de la aplicación el lugar exacto en el que encontrarlo.

De esta forma, el usuario no tiene que recorrer todas las estanterías buscando códigos sino que directamente podría dirigirse a la estantería concreta para recoger el libro buscado.

6.2.4 RED SOCIAL PARA LOCALIZACIÓN DE AMIGOS

En la versión actual, la búsqueda de personas permite obtener la localización de los despachos que constan en el directorio como lugares de trabajo de una persona en concreto.

La opción de crear una red social a la que se acceda con el usuario y contraseña corporativos permitiría que cada usuario pudiera agregar a una serie de amigos. Estos amigos serían los únicos usuarios capaces de consultar la ubicación en tiempo real del usuario.

De esta forma, es más sencillo que tus amigos te encuentren cuando estás moviéndote por la universidad.

6.2.5 GEOLOCALIZACIÓN DE SERVICIOS UNIVERSITARIOS

Sería interesante poder buscar los servicios universitarios por su nombre ya que es más cómodo buscar utilizando la palabra PIC que utilizando el código 2.0.A.06.

6.2.6 GEOLOCALIZACIÓN DE DESFIBRILADORES

Durante este curso 2011/2012 se han instalado en los diferentes campus una serie de desfibriladores que permiten en caso de emergencia reanimar a una persona.

Teniendo en cuenta la facilidad con la que se pueden georreferenciar estos desfibriladores y su importancia a la hora de salvar vidas, se cree conveniente que en un

futuro próximo se georreferencien para poder ser encontrados con la mayor celeridad posible.

6.2.7 CÓMO LLEGAR

Durante el desarrollo de la aplicación ha surgido la duda sobre cómo mostrar rutas que guíen al usuario hacia el lugar buscado desde su posición actual.

De momento esta funcionalidad se deja como posible mejora ya que actualmente no obtenemos una ubicación muy precisa del usuario.

6.2.8 INFORMACIÓN DE AULAS

En la vista de detalle de lugar, en el caso de las aulas sería muy útil que apareciera información acerca de la ocupación en tiempo real en caso de aulas informáticas y del horario de clases de manera que un profesor pueda saber de inmediato si es posible reservarla o no.

BIBLIOGRAFÍA

- [1] Google Inc. Maps. Aplicación Android, 2012. Disponible en Google Play:
<<https://play.google.com/store/apps/details?id=com.google.android.apps.maps>>.
- [2] McCLENDON, B. A new frontier for Google Maps: mapping the indoors. Official Google Blog, Noviembre 2011. Disponible online en
<<http://googleblog.blogspot.com.es/2011/11/new-frontier-for-google-maps-mapping.html>>.
- [3] Point Inside. Point Inside shopping & travel. Aplicación Android, 2011. Disponible en Google Play:
<<https://play.google.com/store/apps/details?id=com.pointinside.android.app>>.
- [4] Accesium Technology S.L. Universidad de Murcia. Aplicación Android, 2011. Disponible en Google Play:
<<https://play.google.com/store/apps/details?id=com.digio>>.
- [5] The University of Alabama. University of Alabama. Aplicación Android, 2012. Disponible en Google Play:
<<https://play.google.com/store/apps/details?id=com.blackboard.android.central.ua>>.
- [6] Ball State University. Ball State University Map App. Aplicación Android, 2010. Disponible en Google Play:
<<https://play.google.com/store/apps/details?id=mapapp.bsu.edu>>.
- [7] MIT. MIT Mobile. Aplicación Android, 2012. Disponible en Google Play:
<<https://play.google.com/store/apps/details?id=edu.mit.mitmobile2>>.
- [8] IEEE. Ieee std 830-1993, recommended practice for software requirements specifications. Institute of Electronic and Electrical Engineers Press. Diciembre 1993.
- [9] Google. Android Design [en línea].
<<http://developer.android.com/design/index.html>> [Consulta: 25 octubre 2011].
- [10] THE APP DATE. Informe Apps primer trimestre 2012 [en línea].
<<http://madrid.theappdate.com/informe-apps/>>
- [11] ALONSO, Juan Carlos; GARCÍA, Javier; AMESCUA, Antonio. “The use of mobile applications in Spanish Universities, current status and future challenges”. En: Actas de la International Conference of Education, Research and Innovation (ICERI), (Madrid, España 14-16 de noviembre de 2011).
- [12] POLE STAR. Indoor Location Solutions [en línea].
<<http://www.polestar.eu/en/nao-campus/indoor-location-technologies.html>> [Consulta: 5 diciembre 2011].
- [13] ERICSSON LABS. Indoor Maps and Positioning [en línea].
<<https://labs.ericsson.com/apis/indoor-maps-and-positioning/>> [Consulta: 5 diciembre 2011].
- [14] SOFTENG. Metodología Scrum [en línea]. <<http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>> [Consulta: 28 mayo 2012].

Bibliografía

- [15] MASÓ, Joan; POMAKIS, Keith; JULIÀ, Núria. OpenGIS Web Map Tile Service Implementation Standard | OGC® [en línea].
<<http://www.opengeospatial.org/standards/wmts>> [Consulta: 11 octubre 2011].
- [16] GRIMES, Shawn. MapKit Overlays – Session 1: Overlay Map [en línea].
<<http://www.shawngrimes.me/2010/12/mapkit-overlays-session-1-overlay-map/>> [Consulta: 13 octubre 2011].
- [17] plusminus (15 noviembre 2007). Android - System Architecture (In Words). Anddev.org. <<http://www.eslweb.org/resources/index.php?topic=256.0>> [Consulta: 15 mayo 2012].
- [18] Android. Installing the SDK [en línea].
<<http://developer.android.com/sdk/installing.html>> [Consulta: 10 octubre 2011].

APÉNDICES

APÉNDICE A PRESUPUESTO DEL PROYECTO

Una vez conocido el esfuerzo necesario para la realización del proyecto, se pasa a detallar el coste del mismo. En primer lugar se computan los gastos asociados al personal que se encarga de la realización de la aplicación. Según el cargo que ocupen su salario varía.

Como personal se ha incluido a Javier Sánchez Hernández como principal desarrollador de la aplicación. También a Javier García Guzmán, ya que es el tutor del proyecto y también se deben computar sus horas de reuniones y ayuda en el desarrollo de la aplicación. Y por último he incluido a Juan Carlos Alonso Durán, porque, aparte de desarrollar el servicio Web, ha aportado consejo y ayuda en el desarrollo, sobre todo al inicio.

El coste total asociado al personal es de 8.025,6 €.

Personal	Categoría	Coste hombre/hora	Dedicación (Horas)	Coste (Euro)
Sánchez Hernández, Javier	Ingeniero Junior	20	352	7.040
García Guzmán, Javier	Ingeniero Senior	36	17,6	633,6
Alonso Durán, Juan Carlos	Ingeniero Junior	20	17,6	352
Hombre hora		76	Total	8.025,6

Tabla 6.1: Coste asociado al personal

También se deben tener en cuenta como gastos asociados a la realización del proyecto las amortizaciones. En este caso las amortizaciones serán referentes a los dos ordenadores utilizados para el desarrollo de la aplicación. Y el dispositivo móvil Google Nexus S, utilizado para llevar a cabo las pruebas físicas (aparte de las realizadas en el AVD).

Se ha considerado un tiempo de vida útil para los ordenadores de 4 años, mientras que para el móvil se han considerado 3 años de tiempo de vida útil.

El coste imputable se ha calculado utilizando los siguientes datos:

- **A** = número de meses desde la fecha de facturación en que el equipo es utilizado
- **B** = periodo de depreciación (48 o 36 meses dependiendo del tipo de dispositivo)
- **C** = coste del equipo (sin IVA)
- **D** = % de uso que se dedica al proyecto

Teniendo en cuenta los anteriores datos, el coste imputable se calcula mediante la siguiente fórmula de cálculo de amortización:

$$A/B * C * D$$

El coste total asociado a las amortizaciones es de 121,43 €.

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable
Ordenador Toshiba Personal Computer, Procesador 2,27 GHz Intel Core i5, Memoria 4 GB 1.066 DDR	700	100	2	48	29,17
Ordenador MacBook Pro, Procesador 2,9 GHz Intel Core i7, Memoria 8 GB 1.600 MHz SDRAM DDR3	1.312,71	100	2	48	54,70
Nexus S	338,14	100	4	36	37,57
Total					121,43

Tabla 6.2: Coste de Amortizaciones

Para poder llevar a cabo la publicación de la aplicación en Google Play se ha tenido que crear un perfil de desarrollador de Android con el siguiente coste:

Licencia	Coste
Licencia Desarrollador	19,73 ³

Tabla 6.3: Coste de licencia

³ El coste original son 25 \$ de EE.UU. La conversión a euro se ha realizado teniendo en cuenta el cambio del día 21 de junio de 2012.

Por último a continuación se muestra el coste final de la aplicación, sumando todos los costes obtenidos anteriormente. Así pues el coste total del proyecto asciende a 8.183,37 euros (ocho mil ciento ochenta y tres euros con treinta y siete céntimos).

Presupuesto	Coste (Euro)
Coste personal	8.025,60
Coste amortizaciones	121,43
Coste licencias	19,73
Coste Total	8.166,76

Tabla 6.4: Coste total

APÉNDICE B MANUAL DE USUARIO

Este manual va dirigido a los usuarios de la aplicación móvil denominada *UC3M Mapas* en su versión para dispositivos con sistema operativo Android.

En la parte inferior de la Figura 6.1 se puede observar el menú principal de la aplicación que se despliega cuando el usuario pulsa el botón de menú de su terminal y que permite acceder a la configuración de las preferencias, a la vista que muestra información acerca de la aplicación y a la ayuda en la que se explica brevemente cómo utilizar la aplicación mediante diagramas sencillos.

En la vista de preferencias se puede configurar tanto el campus como en el que se busca como el tipo de plano a mostrar. Los tipos de plano disponibles son: normal y satélite y los campus disponibles son como se muestra en la Figura 6.2: Leganés, Getafe y Colmenarejo.



Figura 6.1: Menú de opciones

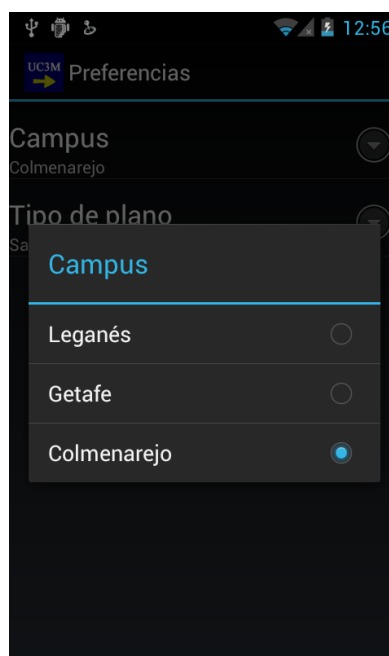


Figura 6.2: Configuración de preferencias

La configuración por defecto de la aplicación corresponde al campus de Leganés, con tipo de mapa satélite y búsqueda de lugar. En la Figura 6.3 se muestra cómo el usuario ha introducido un término de búsqueda en la barra de búsqueda. Además, aparece la posición actual del usuario señalada por un punto azul rodeado de una zona circular semitransparente que indica el posible error en la detección de la posición. En la Figura 6.4 se muestra el resultado de la búsqueda anterior, en el que se muestra el plano del edificio superpuesto, un marcador de posición y una burbuja de información con un botón a través del cual es posible acceder a los detalles del lugar.

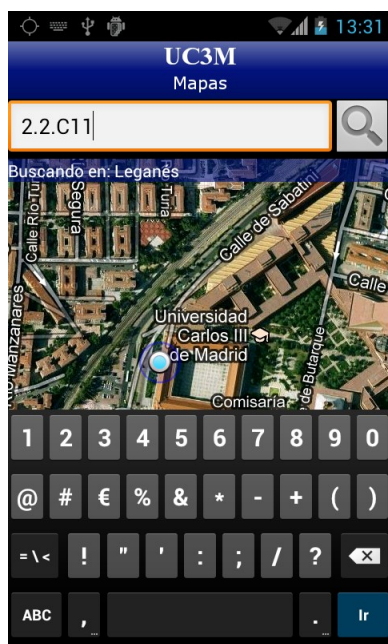


Figura 6.3: Búsqueda de un lugar



Figura 6.4: Resultado de búsqueda de un lugar

Pulsando sobre el botón de la parte inferior de la pantalla con el texto Persona, el tipo de búsqueda cambia a persona y de este modo se puede proceder a buscar a personas en el campus elegido.

En la Figura 6.5 se muestra cómo el usuario ha introducido el término de búsqueda en la barra de búsqueda y en la Figura 6.6 el resultado de tal búsqueda.

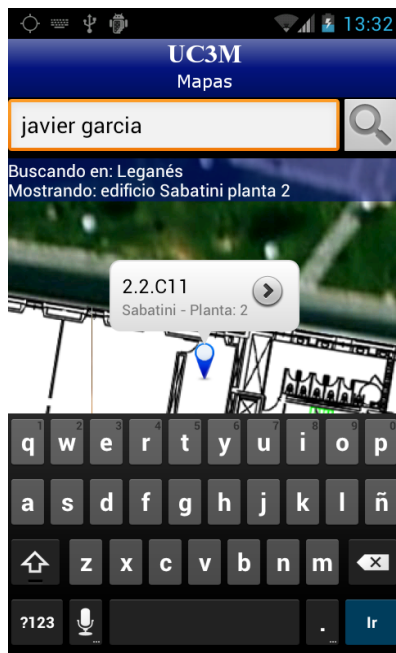


Figura 6.5: Búsqueda de una persona



Figura 6.6: Resultado de búsqueda de una persona

Cuando el usuario pulsa en el botón que se incluye en la burbuja de información, se accede a la vista de detalle de la persona desde la que se puede enviar un e-mail a la persona, llamarle por teléfono u otras opciones. Esta vista de detalle está representada en la Figura 6.7 y en la Figura 6.8 se muestra la opción de agregar a contactos a la persona o de reportar una incidencia al administrador en caso de que la ubicación no sea correcta.



Figura 6.7: Detalle de una persona

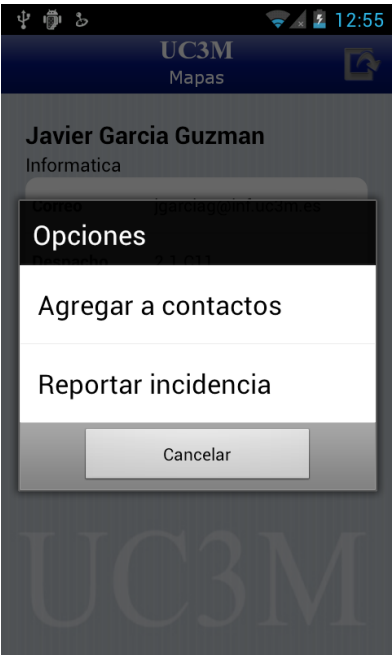


Figura 6.8: Opciones de persona